RNPS No. 2143
ISSN 2072-6260
Versión Digital

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

Minería de subgrafos frecuentes aproximados cerrados en colecciones de multi-grafos

Niusvel Acosta-Mendoza, Andrés Gago-Alonso, José E. Medina-Pagola, Jesús A. Carrasco-Ochoa y José Fco. Martínez-Trinidad

RT_038 **enero 2017**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2143
ISSN 2072-6260
Versión Digital

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

Minería de subgrafos frecuentes aproximados cerrados en colecciones de multi-grafos

Niusvel Acosta-Mendoza, Andrés Gago-Alonso, José E. Medina-Pagola, Jesús A. Carrasco-Ochoa y José Fco. Martínez-Trinidad

RT_038 **enero 2017**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Tabla de contenido

Lista de algoritmos

Minería de subgrafos frecuentes aproximados cerrados en colecciones de multi-grafos

Niusvel Acosta-Mendoza[1,2], Andrés Gago-Alonso[1], José E. Medina-Pagola[1], Jesús A. Carrasco-Ochoa[2], y José Fco. Martínez-Trinidad[2]

[1]Equipo de Investigaciones de Minería de Datos, CENATAV - DATYS, La Habana, Cuba.
{nacosta,agago,jmedina}@cenatav.co.cu
[2]Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México.
{nacosta,ariel,fmartine}@ccc.inaoep.mx

RT_038, Serie Gris, CENATAV - DATYS
Aceptado: 16 de diciembre de 2016

Resumen. En la actualidad se ha incrementado el uso de la minería de subgrafos frecuentes aproximados (SFAs) en diferentes aplicaciones como la clasificación de imágenes, análisis de redes sociales y procesamiento de lenguaje natural, entre otros. En varias de estas aplicaciones, en los últimos años, los multi-grafos han sido utilizados para modelar los datos y solo se han reportado algunos trabajos que permiten minar SFAs en colecciones de multi-grafos [1,2]. Sin embargo, cuando un algoritmo para la minería de SFAs es aplicado, es común que se identifique un gran número de patrones. Para reducir la dimensión del conjunto de patrones manteniendo la eficacia, en este reporte técnico se propone un algoritmo para la minería de SFA cerrados en colecciones de multi-grafos. El comportamiento del algoritmo propuesto es evaluado y comparado con trabajos reportados en la literatura sobre colecciones de multi-grafos. Además, se presenta un experimento que muestra la escalabilidad del algoritmo propuesto.

Palabras clave: minería de multi-grafos, subgrafos frecuentes aproximados cerrados, minería de subgrafos frecuentes aproximados cerrados.

Abstract. Currently, there has been an increment on the use of frequent approximate subgraph (FAS) mining for different real-world applications like image classification, social network analysis and natural language processing, among others. In several of these applications, in the last years, the multi-graphs have been used to model data and only a few works were reported for mining FASs in multi-graph collections [1,2]. However, when a FAS miner is applied, usually a large number of patterns is obtained. Therefore, in order to reducing the dimensionality of the whole set of FASs but keeping the efficacy, in this technical report, we propose an algorithm for mining closed FASs in multi-graph collections. The performance of the proposed algorithm is evaluated over multi-graph collections and it is compared with other works reported in the literature. Besides, an experiment for showing the scalability of the proposal is included.

Keywords: multi-graph mining, closed frequent approximate subgraphs, closed frequent approximate subgraph mining.

1. Introducción

En la Minería de Datos, la minería de patrones frecuentes se ha convertido en un tópico importante con gran aplicación en varios dominios de la ciencia, tales como: la biología, química, ciencias sociales y

lingüísticas, entre otros [3,4,5,6,7,8]. Este tópico incluye diferentes técnicas para la minería de patrones frecuentes, donde se han destacado las de minería de subgrafos frecuentes. Estas técnicas se enfocan en la búsqueda de subgrafos que aparezcan frecuentemente en una base de datos de grafos, donde la frecuencia es acotada mediante un parámetro. Los grafos son utilizados comúnmente para modelar los datos, dado que de forma natural en las aplicaciones existen entidades que pueden representar vértices y las relaciones entre estos pueden ser representadas como aristas [4,5,9,10,11,12,13,14,15,16,17].

Varios algoritmos han sido desarrollados para la minería de subgrafos frecuentes en colecciones de grafos [18,19,20,21,22,23,24,25,26,27,28,29,30]. Estos algoritmos utilizan correspondencias exactas para el cálculo de los subgrafos frecuentes, pero existen varias aplicaciones donde se permiten algunas variaciones en los datos, por ejemplo: análisis redes sociales, de vínculos y de entregas, y clasificación de imágenes, entre otros [5,6,7,31,32,33]. En estas aplicaciones, la correspondencia exacta no produce resultados positivos [7,31,34,35,36,37,38,39,40,41]. Por este motivo, varios algoritmos han sido desarrollados para la minería de subgrafos frecuentes aproximados (SFAs), los cuales utilizan diferentes correspondencias aproximadas entre grafos [35,37,38,39,40,41,42,43,44].

Los algoritmos para la minería de subgrafos frecuentes aproximados se han convertido en importantes herramientas para diferentes aplicaciones, tales como: análisis de estructuras bioquímicas [40,42,45,46], análisis de redes genéticas [47], análisis de circuitos, vínculos y redes sociales [31,41], y clasificación de imágenes [35,38,39]. En algunas de estas aplicaciones existe más de una relación entre pares de vértices, produciendo una representación de multi-grafo[1]. Un ejemplo de esta situación puede observarse en análisis de redes sociales, donde las entidades (personas, videos, objetos, etc.) pueden ser modelados como vértices y las aristas múltiples (multi-aristas) pueden representar las diferentes interacciones entre las entidades [48,49,50,51,52]. Otras redes como las de transporte, rutas, ferroviarias y de travesías pueden ser modeladas con multi-grafos para determinar el costo mínimo de entregas [53] mediante la predicción de contactos entre estaciones [54] o encontrando el camino más barato para viajar [55], entre otros [56,57]. De igual forma, varios trabajos utilizan multi-grafos para representar imágenes en diferentes aplicaciones [10,58,59,60]. En estos trabajos, los autores argumentan que la naturaleza del problema puede ser mejor modelada utilizando multi-grafos en lugar de grafos simples.

Sin embargo, la representación en multi-grafos no ha sido explotada apropiadamente por la carencia de algoritmos que minen los SFAs en colecciones de mutli-grafos. Hasta nuestro conocimiento, solo se han reportado dos trabajos que permiten minar SFAs en colecciones de multi-grafos [1,2]. Sin embargo, es común identificar un gran número de SFAs cuando estos algoritmos son utilizados siendo un reto la identificación de subgrafos interesantes en todo el conjunto de SFAs [3,5,6]. Con el objetivo de solucionar este problema se han propuesto varias técnicas para reducir la dimensión del conjunto de subgrafos, por ejemplo: calcular solo los subgrafos maximales[2], solo los subgrafos cliques[3], o solo los subgrafos cerrados[4], entre otros [16,32,40,61,62,63,64,65,66,67]. En este reporte se propone un nuevo algoritmo para la minería de subgrafos frecuentes aproximados cerrados en colecciones de multi-grafos.

Este trabajo está organizado de la siguiente manera. En la sección 2 se presentan algunos conceptos básicos y se define el problema de la minería de SFA cerrados. En la sección 3 se mencionan los trabajos relacionados. El algoritmo propuesto en este trabajo se describe en la sección 4. Los resultados alcanzados sobre varias colecciones de grafos utilizando la minería de SFA cerrados se detallan en la sección 5. Finalmente, las conclusiones de este trabajo y algunas ideas de trabajo futuros son expuestas en la sección 6.

[1] Un multi-grafo es un grafo que puede contener más de una arista entre un par de vértices (multi-arista), así como aristas que conectan un vértice consigo mismo (lazos).

[2] Un subgrafo maximal es un subgrafo frecuente que no es sub-isomorfo a ningún otro subgrafo frecuente.

[3] Un subgrafo clique es un subgrafo frecuente donde todo vértice está conectado con el resto de los vértices mediante una arista.

[4] Un subgrafo cerrado es un subgrafo frecuente que no es sub-isomorfo a ningún otro con la misma frecuencia.

2. Conceptos básicos

En esta sección se presentan varios conceptos básicos requeridos para definir el problema de la minería de subgrafos frecuentes aproximados (SFAs) cerrados.

Este trabajo está enfocado al trabajo con colecciones de multi-grafos etiquetados y no dirigidos, por lo que los primeros conceptos a definir son los de grafo etiquetado, grafo simple y multi-grafo.

Definición 1 (Grafo etiquetado). *Sean L_V y L_E dos conjuntos de etiquetas para los vértices y las aristas, respectivamente, un grafo etiquetado G es una 5-tupla $(V_G, E_G, \phi_G, I_G, J_G)$ donde:*

- *V_G es un conjunto de vértices,*
- *E_G es un conjunto de aristas,*
- *$\phi_G : E_G \to V_G^{\bullet}$ es una función que devuelve el par de vértices de V_G que están conectados por la arista dada, donde $V_G^{\bullet} = \{\{u,v\} | u,v \in V_G\}$,*
- *$I_G : V_G \to L_V$ es una función etiquetadora para asignar etiquetas a los vértices en V_G,*
- *$J_G : E_G \to L_E$ es una función etiquetadora para asignar etiquetas a las aristas en E_G.*

En la figura 1 se muestra un grafo etiquetado G con $V_G = \{v_0, v_1, v_2\}$ y $E_G = \{e_0, e_1, e_2\}$. En este ejemplo, de acuerdo con la definición 1, $\phi_G(e_0) = \{v_0, v_2\}$, $\phi_G(e_1) = \{v_0, v_1\}$ y $\phi_G(e_2) = \{v_1, v_2\}$, así como $I_G(v_0) = A$, $I_G(v_1) = B$, $I_G(v_2) = C$, $J_G(e_0) = 0$, $J_G(e_1) = 2$ y $J_G(e_2) = 1$.

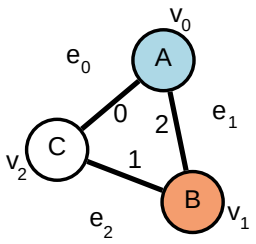
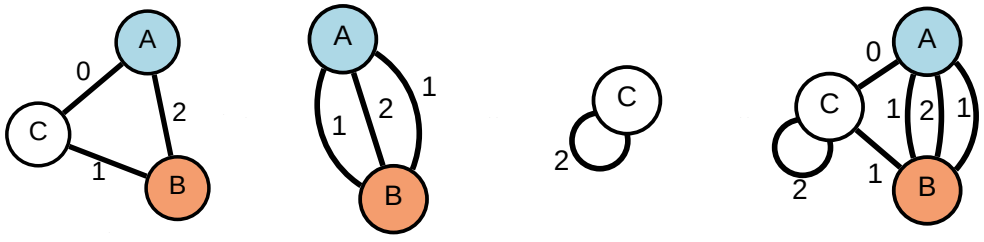


Fig. 1. Ejemplo de un grafo etiquetado G, donde $V_G = \{v_0, v_1, v_2\}$, $E_G = \{e_0, e_1, e_2\}$, $L_V = \{A, B, C\}$ y $L_E = \{0, 1, 2\}$.

Como este trabajo está enfocado en grafos no dirigidos y etiquetados, el dominio de todas las posibles etiquetas es denotado como $L = L_V \cup L_E$. En lo adelante, cuando nos refiramos a un grafo se asumirá un grafo no dirigido y etiquetado. En la figura 2(a), se muestra un ejemplo de un grafo no dirigido y etiquetado con $L_V = \{A, B, C\}$ y $L_E = \{0, 1, 2\}$.



(a) Un grafo. (b) Multi-aristas. (c) Un lazo. (d) Un multi-grafo.

Fig. 2. Ejemplo de diferentes tipos de grafos no dirigidos y etiquetados con $L_V = \{A, B, C\}$ y $L_E = \{0, 1, 2\}$.

Las multi-aristas, como se muestra en la figura 2(b), son diferentes aristas que conectan el mismo par de vértices (i.e. e y e' son multi-aristas si $e \neq e'$ y $\phi_G(e) = \phi_G(e') = \{u,v\}$ tal que $u,v \in V_G$, $u \neq v$). Un lazo, como se puede observar en la figura 2(c), es una arista que conecta un vértice consigo mismo (i.e., cuando $\phi_G(e) = \{u\}$ dado que $\phi_G(e) = \{u,v\}$ con $v = u$; en un lazo $|\phi_G(e)| = 1$).Entonces, un multi-grafo es un grafo que puede contener más de una arista entre un par de vértices (multi-aristas) y aristas conectando un vértice consigo mismo (lazos). En la figura 2(d) se muestra un ejemplo de multi-grafo, donde existen multi-aristas conectando los vértices A y B, y el vértice C tiene un lazo. Luego, a continuación se definen los conceptos de grafo simple y multi-grafo.

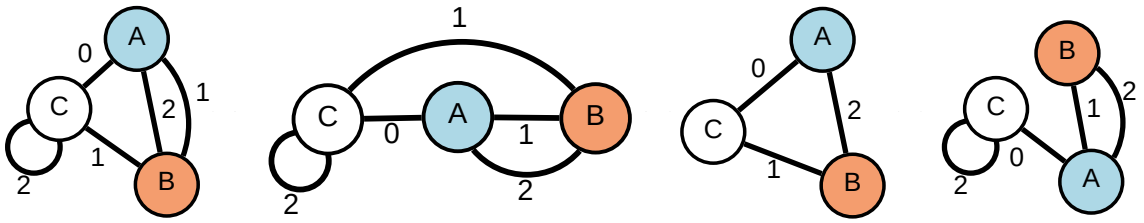
Definición 2 (Grafo simple y multi-grafo). *Un grafo G es un* grafo simple *si no contiene lazos ni multi-aristas; en otro caso, G es un* multi-grafo.

En la minería de grafos exacta, la correspondencia entre grafos se realiza mediante el isomorfismo entre grafos. Para ambos, grafos simples y multi-grafos, el isomorfismo y sub-isomorfismo entre dos grafos son definidos como sigue:

Definición 3 (Isomorfismo y sub-isomorfismo). *Dados dos grafos $G_1 = (V_{G_1}, E_{G_1}, \phi_{G_1}, I_{G_1}, J_{G_1})$ y $G_2 = (V_{G_2}, E_{G_2}, \phi_{G_2}, I_{G_2}, J_{G_2})$, el par de funciones (f,g) es un isomorfismo entre estos grafos si y solo si $f : V_{G_1} \to V_{G_2}$ y $g : E_{G_1} \to E_{G_2}$ son funciones biyectivas, tal que:*

- $\forall u \in V_{G_1} : f(u) \in V_{G_2}$ *y* $I_{G_1}(u) = I_{G_2}(f(u))$
- $\forall e_1 \in E_{G_1}$, *donde* $\phi_{G_1}(e_1) = \{u,v\}$*:* $e_2 = g(e_1) \in E_{G_2}$, *y* $\phi_{G_2}(e_2) = \{f(u), f(v)\}$ *y* $J_{G_1}(e_1) = J_{G_2}(e_2)$.
- $\forall e_1 \in E_{G_1}$, *donde* $\phi_{G_1}(e_1) = \{v\}$*:* $e_2 = g(e_1) \in E_{G_2}$, *y* $\phi_{G_2}(e_2) = \{f(v)\}$ *y* $J_{G_1}(e_1) = J_{G_2}(e_2)$.

Si existe un isomorfismo entre G_1 y G_2, entonces se dice que G_1 y G_2 son isomorfos. Además, si G_1 es isomorfo a un subgrafo de G_2, entonces existe un sub-isomorfismo entre G_1 y G_2; en este caso se dice que G_1 y G_2 son sub-isomorfos (ver figura 3).



(a) Un multi-grafo G_1. (b) Un multi-grafo G_2. (c) Un grafo simple G_3. (d) Un multi-grafo G_4.

Fig. 3. Ejemplo de tres multi-grafos y un grafo simple, donde existe un isomorfismo entre G_1 y G_2, y G_3 y G_4 son sub-isomorfos a G_1 y G_2.

Los algoritmos para la minería exacta de grafos utiliza el isomorfismo (ver definición 3) entre grafos para realizar la correspondencia entre estos. Sin embargo, en ocasiones las colecciones de grafos contienen ruido y los grafos tienen pequeñas variaciones en los vértices y aristas, así como en sus etiquetas. Por este motivo, trabajar con cierta flexibilidad en la correspondencia entre grafos, teniendo en cuenta el ruido de las colecciones de grafos, permite identificar patrones que en otro caso no se encontrarían [38,44,68,69]. Por este motivo, este trabajo está enfocado en la minería de grafos basada en la *correspondencia aproximada entre grafos*, la cual consiste en identificar grafos que son similares pero no idénticos. En la literatura

se han reportado varios trabajos enfocados en este tipo de minería basados en diferentes correspondencias entre grafos, como son: distancia de edición [31,38,39], homeomorfismo [46,70], sustituciones entre etiquetas [35,42,69], entre otros [36,71,72,73,74]. La mayoría de estos trabajos permiten variaciones en las etiquetas de los vértices y las aristas, así como variaciones en la estructura de los grafos. Sin embargo, permitir estas variaciones incrementa considerablemente el costo computacional de los algoritmos para la minería de subgrafos frecuentes. Esto ocurre porque cuando las sustituciones entre etiquetas son combinadas con las variaciones estructurales del grafo, el espacio de búsqueda de la minería aproximada de grafos incrementa considerablemente debido a la explosión combinatoria del número de candidatos y sus ocurrencias. Por esta razón, en este trabajo solo se permitirán variaciones en las etiquetas de los vértices y aristas manteniendo la estructura del grafo. En este escenario se requiere una función de semejanza que permita realizar comparaciones aproximadas entre las etiquetas de los grafos sin afectar la estructura de los mismos. Por lo que se introduce la siguiente definición.

Definición 4 (Semejanza entre las etiquetas de los grafos manteniendo la estructura). *Sean G_1 y G_2 dos multi-grafos, donde V_{G_1}, E_{G_1}, V_{G_2}, y E_{G_2} son sus conjuntos de vértices y aristas, respectivamente. La semejanza entre G_1 y G_2, manteniendo la estructura del grafo es definida como:*

$$sim(G_1,G_2) = \begin{cases} \max\limits_{(f,g)\in\Upsilon(G_1,G_2)} \Theta_{(f,g)}(G_1,G_2) & if\ \Upsilon(G_1,G_2) \neq \emptyset \\ 0 & en\ otro\ caso \end{cases}, \quad (1)$$

donde $\Upsilon(G_1,G_2)$ es el conjunto de todos los posibles isomorfismos entre G_1 y G_2 sin tener en cuenta las etiquetas, y $\Theta_{(f,g)}(G_1,G_2)$ es la función de semejanza para comparar la información de las etiquetas de G_1 y G_2, según el isomorfismo (f,g).

La función de semejanza $\Theta_{(f,g)}$ puede ser definida mediante diferentes operaciones en las etiquetas de los vértices y las aristas, por ejemplo: $\Theta_{(f,g)}$ pudiera ser definida como el producto de los valores de semejanza entre etiquetas. En este caso, si se utiliza esta $\Theta_{(f,g)}$, considerando los dos multi-grafos (G_1 y G_2) que se muestran en la figura 4, donde las etiquetas *A*, *C*, y 1 pueden remplazar las etiquetas *C*, *B*, y 2 con una semejanza de 0,7, 0,6, y 0,8 respectivamente. Entonces, si se aplica una semejanza basada en la correspondencia exacta, G_1 y G_2 no son similares; mientras que si se aplica una semejanza basada en la correspondencia aproximada como la definida antes (ver definición 4), G_1 y G_2 son semejantes con $sim(G_2,G_1) = 0{,}336$.

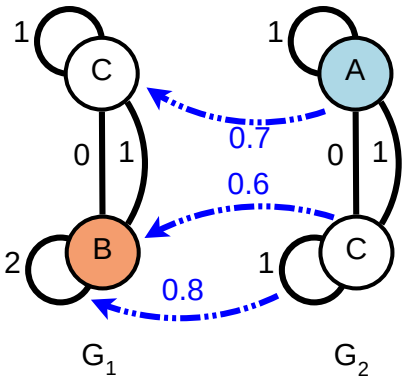


Fig. 4. Ejemplo de correspondencia entre dos multi-grafos G_1 y G_2, donde la etiqueta 2 puede ser reemplazada por la etiqueta 1 con una semejanza de 0,8, la etiqueta *B* puede ser reemplazada por la etiqueta *C* con una semejanza de 0,6 y la etiqueta *C* puede ser reemplazada por la etiqueta *A* con una semejanza de 0,7.

Basado en el concepto de semejanza entre grafos, se utiliza la definición 5 para calcular el isomorfismo y el sub-isomorfismo entre multi-grafos en un contexto de correspondencia aproximada entre grafos.

Definición 5 (Isomorfismo aproximado y sub-isomorfismo aproximado). *Sean G_1, G_2 y G_3 tres multi-grafos, sea $sim(G_1,G_2)$ una función de semejanza que preserva la estructura del grafo, y sea $\tau \in [0,1]$ un umbral de semejanza, existe un isomorfismo aproximado entre G_1 y G_2 si $sim(G_1,G_2) \geq \tau$. Además, si existe un isomorfismo aproximado entre G_1 y G_2, y G_2 es un subgrafo de G_3, entonces existe un sub-isomorfismo aproximado entre G_1 y G_3, denotado como $G_1 \subseteq_A G_3$.*

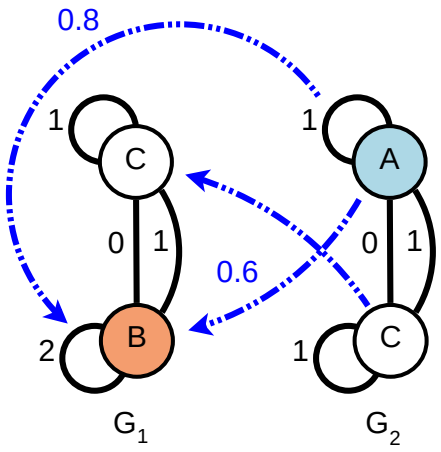


Fig. 5. Ejemplo de correspondencia entre dos multi-grafos G_1 y G_2, donde la etiqueta 2 puede ser reemplazada por la etiqueta 1 con una semejanza de 0,8 y la etiqueta *B* puede ser reemplazada por la etiqueta *A* con una semejanza de 0,6.

En las figuras 4 y 5 se muestran dos formas diferentes de calcular la semejanza aproximada entre el mismo par de multi-grafos (G_1 y G_2). Suponiendo que $\tau = 0{,}3$ y $\Theta_{(f,g)}$ es el mismo del ejemplo anterior, ambas semejanzas $sim(G_2,G_1) = 0{,}336$ (ver figura 4) y $sim_2(G_2,G_1) = 0{,}48$ (ver figura 5) cumplen con el umbral de semejanza τ. Por lo tanto, según la definición 5, estas semejanzas pueden ser usadas para calcular dos diferentes isomorfismos aproximados entre G_1 y G_2. Como se puede notar, entre dos multi-grafos, se pueden calcular más de una semejanza aproximada con diferentes valores. Por lo tanto, se utiliza la siguiente definición con el objetivo de tener solo un valor de semejanza entre dos grafos.

Definición 6 (Grado máximo de inclusión). *Sean G_1 y G_2 dos multi-grafos, sea $sim(G_1,G_2)$ una función de semejanza manteniendo la estructura de los grafos; el grado máximo de inclusión de G_1 en G_2 se define como:*

$$maxID(G_1,G_2) = \max_{G \subseteq G_2} sim(G_1,G), \tag{2}$$

donde $maxID(G_1,G_2)$ significa el valor máximo de semejanza al comparar G_1 con todos los subgrafos de G_2.

Regresando al ejemplo anterior (ver figuras 4 y 5), suponiendo que estas figuras muestran todas las posibles correspondencias para calcular las semejanzas entre G_1 y G_2, el grado de máxima inclusión es 0,48 porque es el valor máximo de semejanza al comparar G_2 con todos los subgrafos de G_1.

Con la definición 7 es posible calcular el soporte aproximado de un subgrafo en una colección. Este soporte aproximado es usado para minar los SFAs en el enfoque aproximado tratado en este trabajo.

Definición 7 (Soporte aproximado). *Sea* $D = \{G_1, \ldots, G_{|D|}\}$ *una colección de multi-grafos, sea* $sim(G_1, G_2)$ *una función de semejanza entre dos multi-grafos, sea* τ *un umbral de semejanza, y sea* G *un multi-grafo. El soporte aproximado (denotado por appSupp) de* G *en* D *se obtiene mediante la ecuación (3):*

$$appSupp(G,D) = \frac{\sum_{G_i \in D, G \subseteq_A G_i} maxID(G, G_i)}{|D|}. \quad (3)$$

Utilizando la ecuación (3) se pueden definir los subgrafos frecuentes aproximados como:

Definición 8 (Subgrafos frecuentes aproximados (SFAs)). *Sea* D *una colección de multi-grafos, sea* G *un multi-grafo y sea* δ *un umbral de soporte,* G *es un subgrafo frecuente aproximado en* D *si y solo si* $appSupp(G,D) \geq \delta$.

Es importante señalar que δ debe tener valores en $[0,1]$ dado que $appSupp(G,D)$ solo tiene valores en ese intervalo.

Teniendo en cuenta la definición de SFA, la *minería de subgrafos frecuentes aproximados* en una colección de multi-grafos consiste en, dado un umbral de soporte, una función de semejanza entre dos multi-grafos y un umbral de semejanza, calcular todos los SFAs en la colección de multi-grafos.

Cuando todos los SFAs de una colección de grafos son comúnmente minados se obtiene un gran número de SFAs. Por esta razón, en la literatura, han sido propuestos algunos tipos de SFAs representativos. Dos de estos tipos de SFAs representativos, conocidos como SFAs maximales y cerrados, son usados para reducir el conjunto de SFAs porque a partir de estos subgrafos se puede reconstruir todo el conjunto de SFAs. Un SFA maximal en una colección de multi-grafos es un SFA que no es sub-isomorfo a ningún otro SFA; mientras que un SFA cerrado en una colección de multi-grafos es un SFA que no es sub-isomorfo a ningún SFA con el mismo soporte aproximado. No obstante, solo a partir de los subgrafos cerrados es posible recuperar la información sobre el soporte de los patrones no cerrados. Por este motivo, el problema tratado en este trabajo es la *minería de SFAs cerrados* en colecciones de multi-grafos, el cual consiste en, dado un umbral de soporte, una función de semejanza entre dos multi-grafos y un umbral de semejanza, calcular todos los SFAs cerrados en la colección de multi-grafos.

3. Trabajos relacionados

En la literatura, los algoritmos para la minería de subgrafos frecuentes han sido desarrollados para trabajar sobre datos representados como un solo grafo [16,31,32,36,37,38,41,45,68,73,74,75,76,77], y colecciones de grafos [21,29,35,39,40,64,78,79,80]. Este trabajo está enfocado en los algoritmos para la minería de subgrafos frecuentes en colecciones de grafos, por lo que nos referiremos a este tipo de algoritmos.

Varios algoritmos para la minería de subgrafos frecuentes en colecciones de grafos han sido propuestos [18,19,20,22,27,29]. Estos algoritmos usan una búsqueda a lo ancho creciendo los subgrafos mediante un vértice o una arista a la vez. Sin embargo, estos algoritmos tienen un alto costo en el proceso de generación de candidatos. Con el objetivo de reducir este costo se han desarrollado otros algoritmos basados en un enfoque de crecimiento de patrones [21,23,24,25,26,28,30,64]. Esos últimos extienden los subgrafos frecuentes adicionando una arista en todas las posibles posiciones. Sin embargo, un problema en este proceso de generación de candidatos es que el mismo subgrafo puede ser obtenido varias veces (i.e. candidatos duplicados). Para reducir la generación de los duplicados, cada subgrafo debe ser extendido de manera conservadora.

Los algoritmos antes mencionados fueron diseñados para minar de forma exacta los subgrafos frecuentes. Sin embargo, en muchas aplicaciones es común que los datos contengan algunas variaciones, causando

que la correspondencia exacta no pueda ser aplicada satisfactoriamente [31,34,35,36,37,38,39,40,41]. Por este motivo, es importante permitir cierto nivel de variabilidad, por ejemplo: variaciones en los vertices y aristas, así como en sus etiquetas. Por esta razón, en el contexto de la minería de grafos se requiere evaluar la semejanza considerando correspondencia aproximada entre los grafos [6,33,81,82]. De esta manera, han sido desarrollados varios algoritmos basados en correspondencias aproximadas para la minería de SFAs [35,39,40,42,43,44,46,47,83]. Diferentes enfoques aproximados han sido usados como base en los algoritmos para la minería de subgrafos frecuentes, por ejemplo: distancia de edición [39,40,47,83], homeomorfismo [46], grafos inciertos [33], y permitiendo solo variaciones en las etiquetas [35,42].

En el enfoque basado en la distancia de edición, diferentes heurísticas basadas en operaciones de edición han sido usadas para comparar grafos con el objetivo de minar SFAs en colecciones de grafos [39,40,47,71,72,83]. Sin embargo, es común que los algoritmos para la minería de SFAs basados en la distancia de edición no minan todos los SFAs, como es el caso de SUBDUECL [83] y FASMGED [39]. En la literatura, solo unos pocos algoritmos, por ejemplo CSMiner [46,70], RAM [71,72] y REAFUM [40], minan todos los SFAs permitiendo variaciones en la estructura de los grafos.

Este trabajo está enfocado en los algoritmos para la minería de SFAs en colecciones de multi-grafos que permiten variaciones en las etiquetas de los vértices y aristas manteniendo la estructura de los grafos. De los algoritmos reportados en la literatura, solo algunos trabajos están basados en este enfoque aproximado [1,2,35,43]. Sin embargo, solo dos trabajos están diseñados para minar SFAs en colecciones de multi-grafos [1,2]. El primero es un método basado en transformaciones de grafos [1], donde una colección de multi-grafos es transformada en una colección de grafos simples, luego se aplica un algoritmo tradicional para la minería de SFAs en la colección de grafos simples, y finalmente los patrones encontrados son retornados al contexto original de multi-grafos. El segundo trabajo introduce una extensión de la forma canónica basada en árboles generados mediante la búsqueda en profundidad para representar multi-grafos, y utilizando esta forma canónica extendida se propone un algoritmo, conocido como AMgMiner, para la minería de SFAs en colecciones de multi-grafos. No obstante, al aplicar estos enfoques se calcula un gran número de SFAs, lo cual hace engorroso el análisis de los patrones obtenidos.

Con el objetivo de reducir el conjunto de SFAs se han desarrollado unos pocos algoritmos para la identificación de SFAs representativos, como es el caso de RNGV [47], que mina los SFAs cerrados, y APGM [42,69] que mina los cliques. Sin embargo, ninguno de estos algoritmos está diseñado para trabajar sobre multi-grafos, además ninguno permite variaciones en las etiquetas de vértices y aristas manteniendo la estructura de los grafos. Por este motivo, en este trabajo se propone un algoritmo para la minería de SFAs cerrados en colecciones de multi-grafos.

4. Minería de subgrafos frecuentes aproximados cerrados

En la literatura se han reportado algoritmos para la minería de patrones cerrados con el objetivo de reducir la duplicidad en el conjunto de subgrafos frecuentes identificados [47,67,78,85,86]. Un subgrafo frecuente cerrado es un patrón que no es subgrafo de ningún otro patrón con el mismo valor del soporte [47,67,78,85,86]. Por este motivo, a partir de los patrones cerrados es posible reconstruir todo el conjunto de subgrafos frecuentes incluyendo la información acerca de sus soportes. Es importante señalar que el número de patrones cerrados es menor que el conjunto de todos los subgrafos frecuentes pero se mantiene la representatividad de los datos con ambos conjuntos. Es por esta razón que este trabajo está enfocado a este tipo de patrones.

Existen varios trabajos acerca de los patrones cerrados [47,67,78,85,86,87,88,89], pero algunos investigadores argumentan que se mantienen las redundancias al utilizar el concepto tradicional de patrón

cerrado. Por lo que proponen otra solución conocida como patrón cerrado tolerante a errores (en inglés: error-tolerant closed pattern), el cual fue denotado como patrón α-cerrado [86,88]. Esta solución permite relajar la condición estricta de cerrado. En este trabajo se considerará que un SFA es un SFA α-cerrado según la definición presentada a continuación.

Definición 9 (Subgrafo frecuente aproximado α-cerrado). *Sea D una colección de multi-grafos, sea δ el umbral de soporte, sea α el umbral de cerrado ($\alpha = [0,1]$), y sea G un multi-grafo frecuente en D, G es un SFA α-cerrado si y solo si no existe G' que sea $G \subset G'$ con $appSup(G',D) \geq max([1-\alpha]appSupp(G,D),\delta)$.*

El algoritmo propuesto para minar los SFAs (closedMgVEAM) primero identifica todos los vértices frecuentes aproximados y luego, comenzando por estos vértices, extiende todos los SFAs adicionando una arista a la vez. En este proceso de extensión, closedMgVEAM primero adiciona los lazos y luego las aristas simples y las múltiples siguiendo una estrategia de crecimiento de patrones para generar los candidatos. En este caso se sigue la estrategia de crecimiento de patrones porque ha sido la que mejores resultados ha reportado en la literatura en el contexto de minería de subgrafos frecuentes. Los SFAs (candidatos) extendidos son representados utilizando una forma canónica basada en matrices de adyacencia (i.e. código CAM), la cual fue extendida y presentada en [84] para permitir representar multi-grafos. Este tipo de representación de candidatos es utilizada con el objetivo de acelerar el proceso de la minería al reducir las pruebas de sub-isomorfismo a la comparación de los códigos CAM. Entonces, la condición de α-cerrado se verifica solo para los candidatos frecuentes, almacenando solo aquellos que cumplen esa condición.

Para permitir variaciones en las etiquetas de los vértices y aristas, closedMgVEAM utiliza matrices de sustitución, la cuales contienen semejanzas que indican cuándo una etiqueta puede ser sustituida por otra (ver definición 10).

Definición 10 (Matriz de sustitución). *Una matriz de sustitución $M = (m_{i,j})$ es una matriz $|L| \times |L|$ indizada por el conjunto de etiquetas L, donde $m_{i,i} > m_{i,j}, \forall j \neq i$. Una celda $m_{i,j}$ $(0 \leq m_{i,j} \leq 1, \sum_j m_{i,j} = 1)$ en M contiene un valor de semejanza que indica si la etiqueta i puede ser reemplazada por la etiqueta j.*

Es importante comentar que una matriz de sustitución puede no ser simétrica porque es posible que una etiqueta i pueda reemplazar otra etiqueta j pero puede ocurrir que j no pueda reemplazar a i.

En closedMgVEAM se utilizan dos matrices de sustitución: una para las etiquetas de los vértices (MV) y otra para las etiquetas de las aristas (ME). Entonces, basado en estas matrices se define la función de semejanza entre grafos de la siguiente manera:

Definición 11 (Función de semejanza $\Theta_{(f,g)}$ basada en matrices de sustitución). *Sean G_1 y G_2 dos multi-grafos, y sean MV y ME dos matrices de sustitución en L_V y L_E, respectivamente. La función de semejanza es definida como:*

$$\Theta_{(f,g)}(G_1,G_2) = \prod_{v \in V_{G_1}} \frac{MV_{I_{G_1}(v),I_{G_2}(f(v))}}{MV_{I_{G_1}(v),I_{G_1}(v)}} * \prod_{e \in E_{G_1}} \frac{ME_{J_{G_1}(e),J_{G_2}(g(e))}}{ME_{J_{G_1}(e),J_{G_1}(e)}}, \tag{4}$$

donde (f,g) es un isomorfismo entre G_1 y G_2.

Utilizando la definición 11 se pueden obtener las ocurrencias (ver definición 12) de cada SFA en la colección de multi-grafos.

Definición 12 (Ocurrencias). *Sean G_1, G_2 y T tres multi-grafos, donde T es un subgrafo de G_2, y sea $sim(G_1,T)$ una función de semejanza según la definición 4, utilizando $\Theta_{(f,g)}(G_1,G_2)$ como en la definición 11; entonces, T es una ocurrencia de G_1 en G_2, utilizando un umbral de semejanza τ, si $sim(G_1,T) \geq \tau$.*

Basado en las definiciona 11 y 12, closedMgVEAM obtiene y almacena todas las ocurrencias de cada subgrafo candidato P_j en una colección de multi-grafos D. Entonces, para crecer a P_j tomando en cuenta sus ocurrencias, solo se recorre el subconjunto de multi-grafos $D_j \subseteq D$ donde P_j tiene al menos una ocurrencia. De esta manera, closedMgVEAM reduce el espacio de búsqueda a D_j porque un SFA solo puede crecerse en un multi-grafo donde exista una ocurrencia.

Con el objetivo de detallar a closedMgVEAM, mediante el algoritmo 1 se presenta el pseudo-código del mismo. Una vez que los vértices frecuentes aproximadas son identificados en la colección de multi-grafos dada D, como se muestra en la línea 1 del algoritmo 1, se sigue la idea de la búsqueda en profundidad (en inglés: depth-first search) para extender recursivamente los vértices frecuentes aproximados. Esta extensión recursiva se realiza con la llamada a la función *"Search"*, la cual solo recorre los multi-grafos $G_i \in D$ que contienen al menos una ocurrencia del SFA a extender. Cuando todos los vértices frecuentes aproximados son extendidos, entonces se tiene como resultado el conjunto de los SFAs α-cerrados F.

Algoritmo 1: $closedMgMiner(D, \tau, \delta, \alpha, F)$

Input: D : una colección de multi-grafos, τ : umbral de semejanza, δ : umbral de soporte, α : umbral de corrado.
Output: F : conjunto de subgrafos frecuentes aproximados α-cerrados.

```
1 C ←Los vértices frecuentes aproximados en D basado en la definición 5;
2 F ← C;
3 NF ← ∅;
4 foreach T ∈ C do
5     D_T ←El conjunto de ocurrencias de T en D basado en la definición 12;
6     Search(T, D_T, D, τ, δ, α, F, NF);
```

La función recursiva (Search) que extiende cada SFA es detallada en el algoritmo 2. En esta función, el conjunto de candidatos se obtiene mediante el llamado a la función *"GenCandidate"*; extendiendo un SFA de todas las posibles posiciones mediante una arista a la vez. Entonces, solo a aquellos candidatos que satisfacen al umbral de soporte se les verifica el cumplimiento de la condición de α-cerrado; manteniendo en el conjunto final F solo los SFAs que cumplen con esta condición (ver líneas 3-5 del algoritmo 2). No obstante, todos los candidatos que son frecuentes y no han sido identificados en pasos anteriores son recursivamente extendidos como se muestra en las líneas 3, y 6–9 del algoritmo 2.

Algoritmo 2: $Search(G, D_G, D, \tau, \delta, \alpha, F, NF)$

Input: G : un subgrafo frecuente aproximado, D_G : conjunto de ocurrencias de G, D : una colección de multi-grafos, τ : umbral de semejanza, δ : umbral de soporte, F : conjunto de subgrafos frecuentes aproximados α-cerrados, NF : un conjunto de multi-grafos.
Output: F : conjunto de subgrafos frecuentes aproximados α-cerrados.

```
1 C ← GenCandidate(G, D_G, D, τ);
2 foreach T ∈ C do
3     if appSupp(T, D) ≥ δ then
4         if δ ≥ [1 − α]appSupp(G, D) or appSup(T, D) ≥ [1 − α]appSupp(G, D) then
5             F ← F \ {G}; NF ← NF ∪ {G}; // basado en la definición 9
6         if T ∉ F and T ∉ NF then
7             F ← F ∪ {T};
8             D_T ←El conjunto de ocurrencias de T en D basado en la definición 12;
9             Search(T, D_T, D, τ, δ, α, F, NF);
```

El objetivo de la función GenCandidate, detallado en el algoritmo 3, es calcular todas las extensiones de un SFA T dado. En esta fase de generación de candidatos se buscan todas las extensiones de T y sus ocurrencias en la colección de multi-grafos D_T. Luego, se calcula el código CAM de todos los subgrafos (candidatos) que satisfacen el umbral de semejanza basado en la definición 11. Finalmente, cada candidato G con su correspondiente código CAM y el valor de semejanza son almacenados como resultados en C.

Algoritmo 3: $GenCandidate(T, D_T, \tau, C)$

Input: T : un subgrafo frecuente aproximado, D_T : conjunto de ocurrencias de T, τ : umbral de semejanza.
Output: C : un conjunto de multi-grafos candidatos.

```
1 O ← {(G, T_e) | G es una extensión de T y T_e es una ocurrencia de G en G_k ∈ D_T};
2 foreach (G, T_e) ∈ O do
3     CAM_G = código CAM de G;
4     sim(G, T_e) se inserta en C utilizando CAM_G como identificador;
```

A continuación, se analiza la complejidad computacional de closedMgVEAM para el peor de los casos, donde todos los multi-grafos de la colección D están completamente conectados, cada multi-grafo $G_i \in D$ tiene el mismo tamaño; es decir: cada G_i tiene n vértices y m aristas, y las etiqutas de los vértices y aristas son idénticas. Entonces lo primero que hace closedMgVEAM es calcular los vértices frecuentes aproximados en D recorriendo todos los vértices de la colección, lo cual es $O(n)$. Luego, se extienden todos los SFAs aplicando la estrategia de búsqueda en profundidad obteniendo el conjunto de candidatos, lo cual es $O(m)$ en el peor de los casos. A cada candidato se le calcula el código CAM realizando, en el peor de los casos, todas las permutaciones entre sus vértices, y este proceso es $O(n!)$. Adicionalmente, en el proceso de extensión de candidatos, se recorren todas las aristas que permitan crecer el candidato, función que se repite m veces por cada arista frecuente aproximada. Es importante señalar que la verificación de la condición de α-cerrado es $O(1)$. Luego, este proceso de crecimiento se realiza recursivamente en un ciclo, donde se realizan $m-1$ llamadas recursivas (excluyendo aquellas aristas que hayan sido recorridas). Entonces, considerando los procedimientos anteriormente mencionados, se puede concluir que la complejidad para el crecimiento de patrones desde un vértice frecuente aproximado, en el peor de los casos, es $O((m)^2(m-1)! + m + n!) = O(mm! + n!)$. Dado que todos estos pasos son realizados para cada vértice frecuente aproximado en cada multi-grafo de D, la complejidad de closedMgVEAM es $O(d(n + n(mm! + n!))) = O(dn(mm! + n!))$, donde $d = |D|$ es el número de multi-grafos en D.

5. Experimentos y resultados

Para evaluar el comportamiento del algoritmo propuesto se generaron 500 imágenes con el generados de imágenes aleatorias de Coenen[5]. Cada imagen de esta colección fue representada como un multi-grafo siguiendo la idea propuesta en [1], obteniéndose una colección de multi-grafos con 21 etiqueta de vértices, 48 etiquetas de aristas, donde el promedio de vértices por grafo es 207 y el promedio de aristas por grafo es 278. Para procesar esta colección se utilizaron las matrices de sustitución reportadas en [35], las cuales fueron sugeridas para este tipo de imágenes. Además, según lo propuesto en ese mismo trabajo, el umbral de semejanza adecuado para estas imágenes es $\tau = 0{,}4$, por lo que se utilizó dicho valor para τ.

Los experimentos fueron realizados en una computadora Core 2-Quad con 8 GB en RAM, sobre el sistema operativo GNU/LINUX. Además, todos los algoritmos están desarrollados en ANSI-C.

[5] www.csc.liv.ac.uk/∼frans/KDD/Software/ImageGenerator/imageGenerator.html

En este experimento se comparan los resultados del algoritmo propuesto con las únicas dos soluciones reportadas en la literatura para minar SFAs en colecciones de multi-grafos [1,2]. Primero se presenta una comparación en términos de tiempo de ejecución, como se puede observar en la figura 6.

En la figura 6 se muestra el tiempo de ejecución en segundos alcanzado por los algoritmos reportados en la literatura (Transf [1] y AMgMiner [2]) y por closedMgVEAM. Como se puede observar en esta figura, el algoritmo propuesto tiene un comportamiento similar en la mayoría de los casos que el algoritmo AMgMiner, pero en todos los casos se observa un mejor comportamiento que el método Transf.

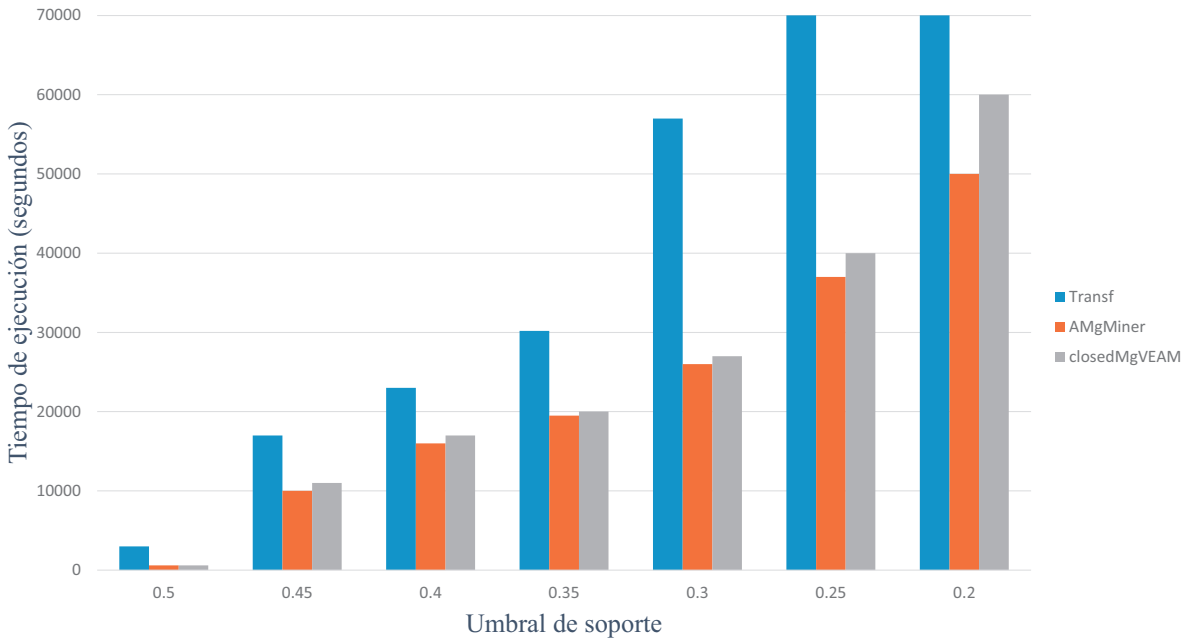


Fig. 6. Tiempo de ejecución del algoritmo propuesto y los reportados en la literatura: Transf [1] y AMgMiner [2].

En la figura 7 se muestra una comparación en términos de cantidad de patrones. Como los algoritmos propuestos en [1,2] encuentran todos los SFAs los resultados de ambos se muestran denotados por "Todos los SFA" en la figura. En este caso se varió el umbral de cerrado α en el algoritmo propuesto para mostrar la variación en las cantidades de patrones que se identifican. Como se puede observar en la figura, la cantidad de patrones cerrados es menor que el conjunto de todos los SFAs, permitiendo representar los datos con un subconjunto de todos los SFAs. Además, se reafirma lo planteado por algunos investigadores sobre los diferentes tipos de patrones cerrados, dado que al aumentar el umbral de cerrado α se reduce el número de patrones. Solo habría que tener en cuenta que esto pudiera afectar a la eficacia de dichos patrones al utilizarlos para representar los datos.

En la figura 8 se muestra una comparación en términos de memoria requerida por los diferentes algoritmos para minar los patrones. Como se puede observar en la figura, la memoria usada por closedMgVEAM es menor que la requerida por AMgMiner en algunos casos, sinedo notablemente menor que la requerida por Transf.

Observando los resultados mostrados en las figuras 6, 7 y 8, se puede concluir que los algoritmos Transf, AMgMiner y closedMgVEAM requieren de mayor tiempo de ejecución a medida que disminuye el umbral de soporte. Esto se debe a que al disminuir dicho umbral, mayor cantidad de subgrafos cumplen con el soporte, lo cual incrementa la cantidad de SFAs. Consecuentemente, es necesario el uso de mayor cantidad de memoria RAM para almacenar la información de los SFAs detectados.

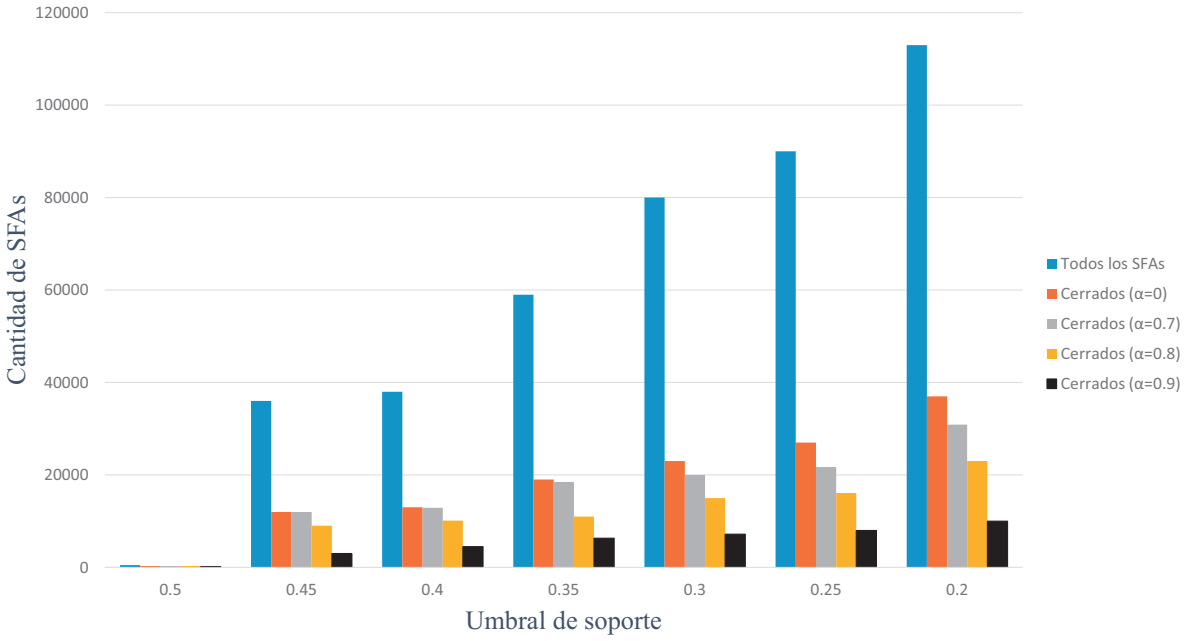


Fig. 7. Cantidad de patrones encontrados.

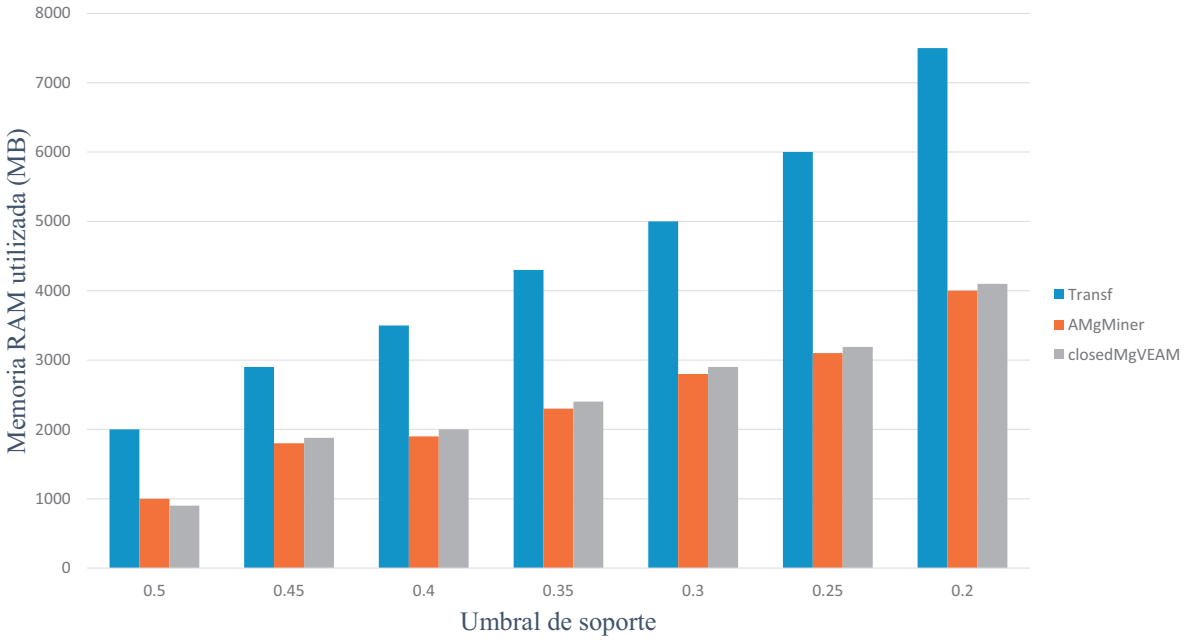


Fig. 8. Memoria RAM utilizada por los algoritmos.

Finalmente, con el objetivo de mostrar la escalabilidad de la propuesta, se incluye un experimento sobre colecciones sintéticas de multi-grafos. En la figura 9 se muestra el tiempo de ejecución de los algoritmos sobre diferentes colecciones de multi-grafos. En el eje horizontal de esta figura se muestra la cantidad de multi-grafos que posee cada colección utilizada. Estas colecciones fueron generadas usando la

libraría generadora de grafos PyGen[6]. Estas colecciones fueron construidas con un máximo de 20 vértices y 50 aristas. En este caso, se fue variando la cantidad de multi-grafos desde 100 hasta 1000 con incremento de 100 y luego se varió esta cantidad de 1000 a 5000 con un incremento de 1000.

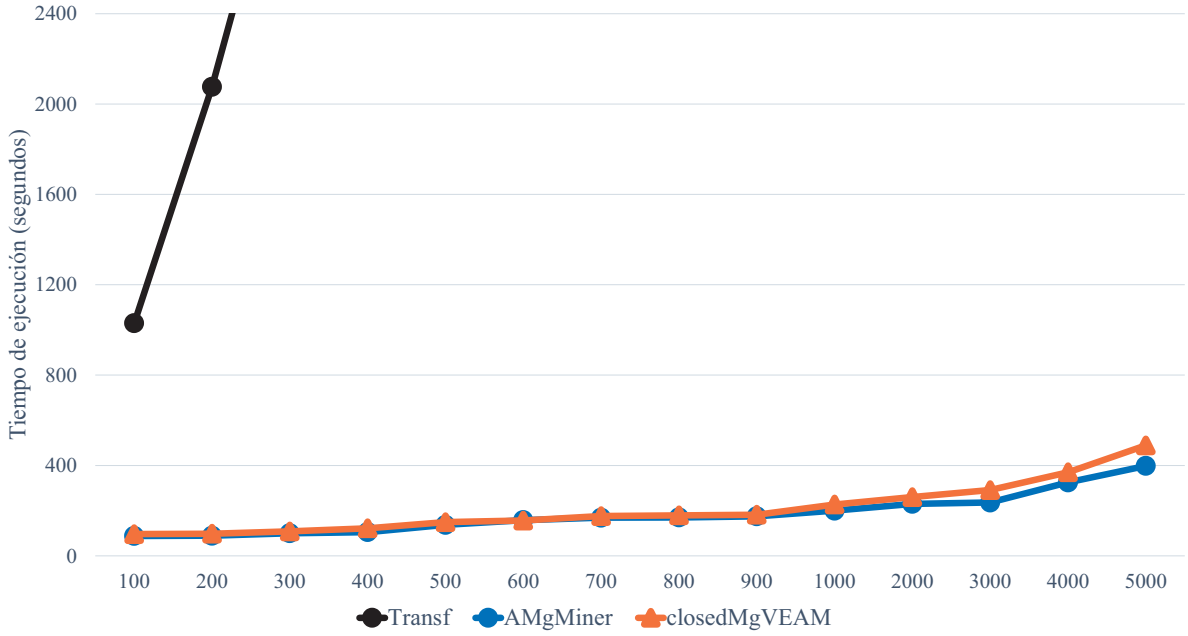


Fig. 9. Tiempo de ejecución de los algoritmos en diferentes colecciones de multi-grafos.

Como se puede observar en la figura 9, una vez más los algoritmos AMgMiner y closedMgVEAM tienen un mejor comportamiento que Transf. Además, estos resultados muestran que el algoritmo propuesto es tan escalable como AMgMiner y que el incremento del número de multi-grafos no afecta drásticamente su comportamiento.

6. Conclusiones y trabajo futuro

Los algoritmos reportados en la literatura minan un conjunto de patrones que el algunos casos no son interesantes o representativos. Estos patrones representativos pueden ser utilizados como entrada para tareas de clasificación o agrupamiento para el descubrimiento de conocimiento en un conjunto de datos y existe la necesidad de algoritmos que permita minar este tipo de patrón representativo sin estar ligado a la naturaleza del problema. Por lo tanto, en este trabajo se propuso un nuevo algoritmo (closedMgVEAM) para la minería de subgrafos frecuentes aproximados cerrados en colecciones de multi-grafos. closedMgVEAM fue descrito detalladamente y su comportamiento fue analizado sobre varias colecciones de multi-grafos y los resultados obtenidos fueron comparados con algoritmos propuestos en la literatura. Basados en los experimentos realizados se puede llegar a la conclusión de que closedMgVEAM permite reducir el conjunto de subgrafos frecuentes aproximados identificados manteniendo la eficiencia reportada en la literatura. Es importante señalar que existe una gran variedad de problemas de investigación relacionados con los patrones cerrados que pudieran ser tratados con el enfoque propuesto en este trabajo. Finalmente, hasta nuestro conocimiento, este es el primer algoritmo reportado en la literatura para la minería de subgrafos

[6] PyGen es una librería que emula grafos (accesible en http://pywebgraph.sourceforge.net).

frecuentes aproximados cerrados en colecciones de multi-grafos que permite variaciones en las etiquetas de vértices y aristas manteniendo la estructura del grafo.

Como trabajo futuro se pretenden proponer algoritmos para minar otros subconjuntos de subgrafos frecuentes aproximados de interés que sean tan o más representativos que los cerrados.

Referencias bibliográficas

1. Acosta-Mendoza, N., Carrasco-Ochoa, J., Martínez-Trinidad, J., Gago-Alonso, A., Medina-Pagola, J.: A New Method Based on Graph Transformation for FAS Mining in Multi-graph Collections. In: The 7th Mexican Conference on Pattern Recognition (MCPR'2015), Pattern Recognition. Volume LNCS 9116. Springer (2015) 13–22
2. Acosta-Mendoza, N., Gago-Alonso, A., Carrasco-Ochoa, J., Martínez-Trinidad, J., Medina-Pagola, J.: A New Algorithm for Approximate Pattern Mining in Multi-graph Collections. Knowledge-Based Systems **109** (2016) 198–207
3. Jiang, C., Coenen, F., Zito, M.: A survey of frequent subgraph mining algorithms. Knowledge Engineering Review (2012)
4. Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A., Rong, X.: Data Mining for the Internet of Things: Literature Review and Challenges. International Journal of Distributed Sensor Networks (2015) 10
5. Ramraj, T., Prabhakar, R.: Frequent subgraph mining algorithms - a survey. Procedia Computer Science **47** (2015) 197–204
6. Emmert-Streib, F., Dehmer, M., Shi, Y.: Fifty years of graph matching, network alignment and network comparison. Information Sciences (2016) 1–22
7. Muñoz-Briseño, A., Lara-Alvarez, G., Gago-Alonso, A., Hernández-Palancar, J.: A Novel Geometric Graph Miner and its Applications. Pattern Recognition Letters (84) (2016) 208–214
8. Wang, K., Xie, X., Jin, H., Yuan, P., Lu, F., Ke, X. In: Frequent Subgraph Mining in Graph Databases Based on MapReduce. Springer International Publishing, Cham (2016) 464–476
9. Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning, Orlando, USA (2008) 208–297
10. Morales-González, A., García-Reyes, E.B.: Simple object recognition based on spatial relations and visual features represented using irregular pyramids. Multimedia tools and applications **63**(3) (2013) 875–897
11. Aoun, N., M.M., Amar, C.: Bag of sub-graphs for video event recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Florence, Italy,. (2014) 1547–1551
12. Patel, J., Oza, B.: Survey on Graph Pattern Mining Approach. International Journal of Engineering Development and Research **2**(1) (2014) 5
13. Manzo, M., Pellino, S., Petrosino, A., Rozza, A.: A novel graph embedding framework for object recognition. In: ECCV 2014 Workshops. Volume Part IV, LNCS 8928., Springer International Publishing Switzerland (2015) 341–352
14. Rousseau, F., Kiagias, E., Vazirgiannis, M.: Text categorization as a graph classification problem. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Volume 1., Beijing, China (2015) 1702–1712
15. Acosta-Mendoza, N., Gago-Alonso, A., Carrasco-Ochoa, J., Martínez-Trinidad, J., Medina-Pagola, J.: Improving Graph-Based Image Classification by using Emerging Patterns as Attributes. Engineering Applications of Artificial Intelligence **50** (2016) 215–225
16. Hao, F., Park, D., Li, S., Lee, H.: Mining λ-Maximal Cliques from a Fuzzy Graph. Sustainability **8**(553) (2016) 1–16
17. Shi, B., Weninger, T.: Discriminative predicate path mining for fact checking in knowledge graphs. Knowledge-Based Systems, DOI: 10.1016/j.knosys.2016.04.015 (2016)
18. Borgelt, C.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proc. IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan, IEEE Press (2002) 51–58
19. Inokuchi, A., Washio, T., Nishimura, K., Motoda, H.: A fast algorithm for mining frequent connected subgraphs. In: Technical Report RT0448, In IBM Research, Tokyo Research Laboratory (2002)
20. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. Technical report, IEEE Transactions on Knowledge and Data Engineering (2002)
21. Yan, X., Huan, J.: gSpan: Graph-Based Substructure Pattern Mining. In: International Conference on Data Mining, Japan, Maebashi (2002)
22. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: The 3rd IEEE International Conference on Data Mining, FL, Melbourne (2003) 549–552
23. Nijssen, S., Kok, J.: A Quickstart in Frequent Structure Mining can make a Difference. In: The 10th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2004) 647–652
24. Wang, C., Wang, W., Pei, J., Zhu, Y., Chi, B.: Scalable Mining of Large Disk-based Graph Databases. In: Proc. of the 2004 ACM SIGKDD of International Conference on Knowledge Discovery in databases (KDD), Seattle, WA (2004) 316–325

25. Zhu, F., Yan, X., Han, J., Yu, P.: gPrune: A Constraint Pushing Framework for Graph Pattern Mining. In: Advances in Knowledge Discovery and Data Mining, 11th Pacific-Asia Conference (PAKDD'07). Volume 4426 of Lecture Notes in Computer Science., Springer (2007) 388–400
26. Gago-Alonso, A., Medina-Pagola, J., Carrasco-Ochoa, J., Trinidad, J.M.: Mining Frequent Connected Subgraphs Reducing the Number of Candidates. In: Machine Learning and Knowledge Discovery in Databases, European Conference, (ECML/PKDD). Volume 5211 of Lecture Notes in Computer Science., Springer (2008) 365–376
27. Thomas, L., Valluri, S., Karlapalem, K.: ISG: Itemset based subgraph mining. Technical Report **IIIT, Hyderabad** (December 2009)
28. Gago-Alonso, A., Carrasco-Ochoa, J.A., Medina-Pagola, J., Martínez-Trinidad, J.: Full Duplicate Candidate Pruning for Frequent Connected Subgraph Mining. Integrated Computer-Aided Engineering **17** (August 2010) 211–225
29. Gago-Alonso, A., Puentes-Luberta, A., Carrasco-Ochoa, J., Medina-Pagola, J., Martínez-Trinidad, J.: A new algorithm for mining frequent connected subgraphs based on adjacency matrices. Intelligent Data Analysis **14** (2010) 385–403
30. Gago-Alonso, A.: Connected Permutations of Vertices for Canonical Form Detection in Graph Mining. Revista Cubana de Ciencias Informáticas **9** (2015) 57–71
31. Holder, L., Cook, D., Bunke, H.: Fuzzy substructure discovery. In: ML92: Proceedings of the ninth international workshop on Machine learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1992) 218–223
32. Flores-Garrido, M., Carrasco-Ochoa, J., Martínez-Trinidad, J.: Mining maximal frequent patterns in a single graph using inexact matching. Knowledge-Based Systems **66** (2014) 166–177
33. Santhi, S., Padmaja, P.: A Survey of Frequent Subgraph Mining algorithms for Uncertain Graph Data. International Research Journal of Engineering and Technology (IRJET) **2**(2) (2015) 688–696
34. Chen, C., Lin, C., Yan, X., Han, J.: On effective presentation of graph patterns: a structural representative approach. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008. (2008) 299–308
35. Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.: Frequent approximate subgraphs as features for graph-based image classification. Knowledge-Based Systems **27** (2012) 381–392
36. Li, J., Zou, Z., Gao, H.: Mining frequent subgraphs over uncertain graph databases under probabilistic semantics. VLDB J. **21**(6) (2012) 753–777
37. Elseidy, M., Abdelhamid, E., Skiadopoulos, S., Kalnis, P.: GRAMI: Frequent Subgraph and Pattern Mining in a Single Large Graph. PVLDB **7**(7) (2014) 517–528
38. Flores-Garrido, M., Carrasco-Ochoa, J., Martínez-Trinidad, J.: AGraP: an algorithm for mining frequent patterns in a single graph using inexact matching. Knowledge and Information Systems (2015) 1–22
39. Gao, L., Pan, H., Han, Q., Xie, X., Zhang, Z., Zhai, X., Li, P.: Finding Frequent Approximate Subgraphs in Medical Image Database. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE (2015) 1004–1007
40. Li, R., Wang, W.: REAFUM: Representative Approximate Frequent Subgraph Mining. In: SIAM International Conference on Data Mining, Vancouver, BC, Canada, SIAM. (2015) 757–765
41. Moussaoui, M., Zaghdoud, M., Akaichi, J. In: POSGRAMI: Possibilistic Frequent Subgraph Mining in a Single Large Graph. Springer International Publishing, Cham (2016) 549–561
42. Jia, Y., Zhang, J., Huan, J.: An efficient graph-mining method for complicated and noisy data with real-world applications. Knowledge and Information Systems **28**(2) (2011) 423–447
43. Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.: On speeding up frequent approximate subgraph mining. In: Proceedings of the 17th Iberoamerican Congress on Pattern Recognition (CIARP'12). Volume LNCS 7441., Buenos Aires, Argentina, Springer-Verlag Berlin Heidelberg (2012) 316–323
44. Morales-González, A., Acosta-Mendoza, N., Gago-Alonso, A., García-Reyes, E., Medina-Pagola, J.: A new proposal for graph-based image classification using frequent approximate subgraphs. Pattern Recognition **47**(1) (2014) 169–177
45. Chen, C., Yan, X., Zhu, F., Han, J.: gApprox: Mining Frequent Approximate Patterns from a Massive Network. In: International Conference on Data Mining (ICDM'07). (2007) 445–450
46. Xiao, Y., Wu, W., Wang, W., He, Z.: Efficient Algorithms for Node Disjoint Subgraph Homeomorphism Determination. In: Proceedings of the 13th international conference on Database systems for advanced applications, New Delhi, India, Springer-Verlag, Berlin, Heidelberg (2008) 452–460
47. Song, Y., Chen, S.S.: Item sets based graph mining algorithm and application in genetic regulatory networks. Data Mining, IEEE International Conference on Volume, Issue (2006) 337–340
48. Jabeur, L., Tamine, L., Boughanem, M.: Active microbloggers: Identifying influencers, leaders and discussers in microblogging networks. In: String Processing and Information Retrieval – 19th International Symposium, Cartagena de Indias, Colombia. Volume LNCS 7608. (2012) 111–117
49. Papalexakis, E., Akoglu, L., Ienco, D.: Do more Views of a Graph help? Community Detection and Clustering in Multi-Graphs. In: 16th International Conference on Information Fusion, IEEE, Istanbul, Turkey. (2013) 899–905

50. Goonetilleke, O., Sathe, S., Sellis, T., Zhang, X.: Microblogging Queries on Graph Databases: An Introspection. In: Third International Workshop on Graph Data Management Experiences and Systems, (GRADES), Melbourne, VIC, Australia. Volume 5. (2015) 1–6
51. Cazabet, R., Takeda, H., Hamasaki, M.: Characterizing the nature of interactions for cooperative creation in online social networks. Social Network Analysis and Mining **5**(1) (2015) 1–17
52. Verma, A., Bharadwaj, K.: Identifying community structure in a multi-relational network employing non-negative tensor factorization and GA k-means clustering. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery (2016)
53. Setak, M., Habibi, M., Karimi, H., Abedzadeh, M.: A time-dependent vehicle routing problem in multigraph with FIFO property. Journal of Manufacturing Systems **35** (2015) 37–45
54. Wang, H., Ma, W., Shi, H., Xia, C.: An Interval Algebra-based Modeling and Routing Method in Bus Delay Tolerant Network. KSII Transactions on Internet and Information Systems **9**(4) (2015) 1376–1391
55. Hulianytsky, L., Pavlenko, A.: Ant Colony Optimization for Time Dependent Shortest Path Problem in Directed Multigraph. International Journal Information Content and Processing **2**(1) (2015) 50–61
56. Terroso-Saez, F., Valdés-Vela, M., Skarmeta-Gómez, A.: Online Urban Mobility Detection Based on Velocity Features. In: 17th International Conference of Big Data Analytics and Knowledge Discovery, Valencia, Spain. Volume LNCS 9263. (2015) 351–362
57. Wei, D., Liu, H., Qin, Y.: Modeling cascade dynamics of railway networks under inclement weather. Transportation Research Part E **80** (2015) 95–122
58. Kropatsch, W., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision pyramids that do not grow too high. Pattern Recognition Letters **26** (2005) 319–337
59. Morales-González, A., García-Reyes, E.B.: Assessing the Role of Spatial Relations for the Object Recognition Task. In: The 15th Iberoamerican Congress on Pattern Recognition (CIARP'10). Volume 6419 of Lecture Notes in Computer Science., Springer, Heidelberg (2010) 549–556
60. Youssef, R., Kacem, A., Sevestre-Ghalila, S., Chappard, C.: Graph Structuring of Skeleton Object for Its HighLevel Exploitation. Image Analysis and Recognition **LNCS 9164** (2015) 419–426
61. Liu, M., Gribskov, M.: MMC-Marging: Identification of Maximum Frequent Subgraphs By Metropolis Monte Carlos Sampling. In: IEEE International Conference on Big Data, IEEE (2015) 849–856
62. Chalupa, D.: On Combinatorial Optimization in Analysis of Protein-Protein Interaction and Protein Folding Networks. Applications of Evolutionary Computation **LNCS 9597** (2016) 91–105
63. Chen, Q., Fang, C., Wang, Z., Suo, B., Li, Z., Ives, Z.: Parallelizing maximal clique enumeration over graph data. Database Systems for Advanced Applications **LNCS 9643** (2016) 249–264
64. El Islem Karabadji, N., Aridhi, S., Seridi, H. In: A Closed Frequent Subgraph Mining Algorithm in Unique Edge Label Graphs. Springer International Publishing, Cham (2016) 43–57
65. Hahn, K., Massopust, P., Prigarin, S.: A new method to measure complexity in binary or weighted networks and applications to functional connectivity in the human brain. BMC Bioinformatics **17**(87) (2016) 1–18
66. Segundo, P., Lopez, A., Pardalos, P.: A new exact maximum clique algorithm for large and massive sparse graphs. Computers and Operations Research **66** (2016) 81–94
67. Salma, M.: An Efficient Algorithm for mining Frequent Pattern Growth without candidate Generation. International Journal of Emerging Trends in Technology and Sciences **06**(03) (2016) 480–487
68. Cook, D.J., Holder, L.B.: Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research **1** (1994) 231–255
69. Jia, Y., Huan, J., Buhr, V., Zhang, J., Carayannopoulos, L.: Towards comprehensive structural motif mining for better fold annotation in the "twilight zone" of sequence dissimilarity. BMC Bioinformatics **10**(S-1) (2009)
70. Xiao, Y., Wang, W., Wu, W.: Mining conserved topological structures from large protein-protein interaction networks. In: Proceedings of the 18th IEICE data engineering workshop / 5th DBSJ annual meeting, Hiroshima, Japan, DEWS'2007 (2007)
71. Zhang, S., Yang, J., Cheedella, V.: Monkey: Approximate Graph Mining Based on Spanning Trees. In: International Conference on Data Engineering, Los Alamitos, CA, USA, IEEE ICDE (2007) 1247–1249
72. Zhang, S., Yang, J.: RAM: Randomized Approximate Graph Mining. In: The 20th International Conference on Scientific and Statistical Database Management, China, Hong Kong (2008) 187–203
73. Zou, Z., Li, J., Gao, H., Zhang, S.: Finding top-k maximal cliques in an uncertain graph. In: IEEE 26th International Conference on Data Engineering (ICDE 2010). (2010) 649–652
74. Zou, Z., Li, J., Gao, H., Zhang, S.: Mining frequent subgraph patterns from uncertain graph data. IEEE Trans. on Knowl. and Data Eng. **22**(9) (2010) 1203–1218
75. Ketkar, N.S.: Subdue: compression-based frequent pattern discovery in graph data. In: OSDM'05: Proceedings of the 1st international workshop on open source data mining, ACM Press (2005) 71–76
76. Thomas, L., Valluri, S., Karlapalem, K.: Margin: Maximal frequent subgraph mining. ACM Trans. Knowl. Discov. Data **4** (October 2010) 10:1–10:42

77. Abdelhamid, E., Abdelaziz, I., Kalnis, P., Khayyat, Z., Jamour, F.: ScaleMine: Scalable Parallel Frequent Subgraph Mining in a Single Large Graph. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '16, Piscataway, NJ, USA, IEEE Press (2016) 61:1–61:12
78. Yan, X., Han, J.: ClosedGraph: Mining Closed Frequent Graph Patterns. In: Proc. of the 9th ACM SIGKDD of International Conference on Knowledge Discovery and Data Mining (KDD), Washington, DC (2003) 286–295
79. Huan, J., Wang, W., Prins, J., Yang, J.: Spin: Mining maximal frequent subgraphs from graph databases. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Minin., ACM (2004) 581–586
80. Chen, X., Zhang, C., Liu, F., Guo, J.: Algorithm research of top-down mining maximal frequent subgraph based on tree structure. In: Snac, P., Ott, M., Seneviratne, A. (Eds.), Wireless Communications and Applications. Volume 72., Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg. (2012) 401–411
81. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI **18**(3) (2004) 265–298
82. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Analysis and Applications **13**(1) (2010) 113–129
83. González, J., Holder, L., Cook, D.: Graph-Based Concept Learning. In: Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference, Key West, Florida, USA, AAAI Press (2001) 377–381
84. Acosta-Mendoza, N., Carrasco-Ochoa, J., Gago-Alonso, A., Martínez-Trinidad, J., Medina-Pagola, J.: Representative Frequent Approximate Subgraph Mining in Multi-Graph Collections. Technical Report Technical Report CCC-15-001, Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico (2015)
85. Borgelt, C., Meinl, T.: Full perfect extension pruning for frequent subgraph mining. Mining Complex Data (2009) 189–205
86. Takigawa, I., Mamitsuka, H.: Efficiently Mining δ-tolerance Closed Frequent Subgraphs. Machine Learning **82**(2) (2011) 95–121
87. Yan, X., Han, J., Afshar, R.: CloSpan: Mining closed sequential patterns in large datasets. In: 3rd SIAM International Conference on Data Mining, San Franciso, USA, SIAM (2003) 166–177
88. Cheng, J., Ke, Y., Ng, W.: δ-Tolerance Closed Frequent Itemsets. In: 6th International Conference on Data Mining (ICDM'06), Hong Kong, IEEE (December 18–22 2006) 139–148
89. Lartillot, O.: Automated Motivic Analysis: An Exhaustive Approach Based on Closed and Cyclic Pattern Mining in Multi-dimensional Parametric Spaces. Computational Music Analysis **Part V** (2015) 273–302

RT_038, enero 2017
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2143
ISSN 2072-6260
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba