

REPORTE TÉCNICO
**Minería
de Datos**

SERIE GRIS

**Predicción de enlaces en redes
sociales mediante grafos
dinámicos utilizando reglas de
asociación**

José Eladio Medina Pagola y
Reynier José Ortiz Tamayo

RT_031

septiembre 2015





CENATAV

Centro de Aplicaciones de
Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2143
ISSN 2072-6260
Versión Digital

REPORTE TÉCNICO
**Minería
de Datos**

SERIE GRIS

**Predicción de enlaces en redes
sociales mediante grafos dinámicos
utilizando reglas de asociaciones**

José Eladio Medina Pagola y
Reynier José Ortiz Tamayo

RT_031

septiembre 2015



Tabla de contenido

1	Introducción	2
2	Conceptos preliminares	4
3	Grafos dinámicos	5
4	Métodos basados en las reglas de evolución de grafos	6
4.1	Reglas de evolución de grafos.....	6
4.1.1	Patrones en las reglas de evolución de grafos	7
4.1.2	Cálculo del soporte.....	8
4.1.3	Cálculo de la confianza	12
4.1.4	Nuevas métricas de calidad de reglas de evolución	13
4.1.5	Algoritmo GERM.....	13
4.1.6	Análisis del método GERM	15
4.2	Método TP-Miner.....	16
4.2.1	Multiplicidad de aristas	16
4.2.2	Cálculo de la confianza	16
4.2.3	Análisis del método.....	17
5	Reglas de formación de enlaces	17
5.1	Estructura del grafo dinámico	18
5.2	Patrones y reglas de formación de enlace	18
5.3	Ocurrencia y frecuencia de un patrón LF	19
5.4	Algoritmo LFR-Miner.....	20
5.5	Análisis del método LFR-Miner	22
6	Consideraciones generales	23
	Conclusiones	24
	Referencias bibliográficas.....	25

Predicción de enlaces en redes sociales mediante grafos dinámicos utilizando reglas de asociación

José Eladio Medina Pagola y Reynier José Ortiz Tamayo

Equipo de Investigaciones en Minería de Datos, Centro de Aplicaciones de Tecnologías de Avanzada
(CENATAV), La Habana, Cuba
{jmedina, rortiz} @cenatav.co.cu

RT_031, Serie Gris, CENATAV
Aceptado: 1 de septiembre de 2015

Resumen. En el campo de la predicción de enlace, los métodos asociados a la minería de grafos dinámicos han experimentado un significativo y creciente desarrollo en los últimos años, acentuado por la necesidad de analizar la evolución de las asociaciones en las redes sociales. Las técnicas relacionadas con este tipo de minería pueden ser vistas como una evolución de la minería de patrones frecuentes en grafos estáticos hacia grafos dinámicos. La minería de reglas de evolución de grafos es uno de los primeros enfoques que considera la evolución de las redes sociales para llevar a cabo la predicción de enlace. A partir de esta se han creado otros estudios que siguen similares criterios con contribuciones significativas. Estos trabajos son analizados en este reporte, presentándose sus principales ventajas y desventajas. También son mencionadas las posibles extensiones y aplicaciones de estas técnicas, en vistas de lograr predicciones de enlace de mayor eficacia y significación.

Palabras clave: Predicción de enlace, minería de subgrafos frecuentes, grafos de evolución, redes sociales.

Abstract. In link prediction field, the methods related to dynamic graph mining have shown a significant and growing development in the last years, marked by the need to analyze the evolution of the associations in social networks. The techniques related to this type of mining can be seeing as an evolution of frequent pattern mining in static graphs to dynamic graphs. Graph evolution rule mining is one of the first approaches that take into account the evolution of social networks for performing link prediction. From this approach, other works have been created following similar ideas with significant contributions. These works are analyzed in this report, presenting its main advantages and disadvantages. Furthermore, with the goal of obtaining links predictions with high effectiveness and significance, possible extensions and applications of these techniques are mentioned.

Keywords: Link prediction, frequent subgraph mining, evolution graphs, social networks.

1 Introducción

En la actualidad se observa la necesidad de definir modelos que permitan describir las interacciones presentes en múltiples escenarios reales. Lo anterior se debe, en gran medida, al alto nivel de interconectividad que existe en el mundo, los grandes volúmenes de información que son generados, así como la complejidad de los sistemas y de sus datos. Una de las formas de organizar y procesar tales datos es considerando las concepciones y métodos asociados con lo conocido como Análisis de Redes Sociales.

El Análisis de las Redes Sociales surge y se formaliza por la necesidad de comprender, visualizar y extraer conocimientos de los datos contenidos en este tipo de red. Se asume como red social aquella que permite representar relaciones e interacciones entre entidades sociales, tales como individuos, grupos o comunidades. Cuando una red social se enfoca a la representación de personas y sus interacciones, las asociaciones son generalmente impulsadas por intereses mutuos entre los usuarios, que son intrínsecos a un grupo. Una característica muy distintiva de las redes sociales es su dinamismo, ya que las entidades y relaciones se modifican en la red conforme pasa el tiempo.

La comprensión de la dinámica que impulsa la evolución de la red social es un problema complejo, debido al gran número de factores que están presentes. En la actualidad con el avance de las tecnologías se hace cada vez más necesaria la aplicación de técnicas matemáticas y computacionales que permitan el descubrimiento de patrones o relaciones en redes sociales, contenidos en diversos tipos y formatos de bases de datos. Cuando estos patrones consideran la identificación de enlaces que pudieran producirse en un futuro, se habla de Predicción de Enlace (*Link Prediction*). La predicción de enlaces puede ser definida como el problema de estimar la probabilidad de creación o desaparición de un enlace entre dos nodos en un futuro cercano o no, basados en enlaces ocurridos en el pasado [38]. También se aplica este término cuando se persigue evaluar la posible ausencia de enlaces en una red para un instante de tiempo, lo cual podría considerarse como una anomalía [52].

Las técnicas de predicción de enlace pudieran ser analizadas bajo diferentes estructuraciones. Algunos especialistas las agrupan atendiendo a si los atributos que consideran son locales al enlace a predecir o globales a toda la red social [26], [28], o considerando una tercera clase denominada como quasi locales [40].

Otros especialistas, aún cuando consideran válida la anterior clasificación, asumen otras posibles estructuraciones atendiendo a los fundamentos teóricos en los que se sustentan. Tal es el caso de Hasan y Zaki, los que agrupan los métodos en aquellos que consideran métricas de semejanza entre pares de nodos, los que se basan en modelos probabilísticos Bayesianos, y los que consideran modelos relacionales probabilísticos [28]. En el primer grupo puede mencionarse los que consideran la métrica de los *Vecinos en Común* (cuanto mayor sea el número de vecinos en común entre dos nodos, mayores serán las posibilidades de que se relacionen estos en un futuro cercano), con extensiones tales como el *Coefficiente de Jaccard*, o la propuesta Adamic/Adam [2], [37]. También en este grupo pueden ser incluidos aquellos que consideran la métrica de los *Caminos Más Cortos* (cuanto menor sea el camino entre dos nodos, mayor será la probabilidad de que se relacionen entre sí). No obstante, debido al fenómeno de *Mundo Pequeño* (dos nodos cualesquiera de una red están conectados por una pequeña cantidad de vértices) varios trabajos han argumentado el porqué esta medida puede resultar engañosa; muestra de ello es lo presentado por Liben-Nowell *et al.* [26] y por Hasan *et al.* [27] en redes de co-autoría. Como una variante de los *Caminos Más Cortos*, Leo Katz propuso una métrica con la cual se han obtenido buenos resultados en la predicción de enlace [33]. Uno de los problemas de esta variante es su costo computacional. Los métodos *Hitting Time* [28] y *Rooted Pagerank* [15] son dos de los trabajos con los cuales también se han logrado buenos resultados considerando los *Caminos más Cortos*.

Otros trabajos que evalúan la semejanza entre nodos son los que consideran los atributos de los vértices y las aristas [20], [27], [28]; tal es el caso de la métrica *Preferential Attachment* [7]. Hasan *et al.* mostraron buenos resultados con este enfoque [27].

Una clasificación más extensa y de mayor actualidad es la realizada por Wang *et al.* [52]. En esta se agrupan las técnicas atendiendo a si: consideran atributos y métricas basadas en los nodos, los que se basan en vecindades, los basados en caminos aleatorios (random walk), y los basados en teorías sociales. Al mismo tiempo, estos autores resaltan como otra clasificación: los basados en las semejanzas de rasgos, los basados en kernels, los basados en factorización de matrices, y los modelos probabilísticos.

Todos estos métodos antes mencionados centran su atención en los patrones estáticos, donde la minería se realiza sobre una instantánea de la red social, o sobre el grafo resultante de la acumulación de toda la evolución de la red. Sin embargo, con el avance de las tecnologías y el constante flujo de información se imponía tener en cuenta la evolución y la dinámica de las redes sociales. Por lo que se han reportado trabajos sobre predicción de enlace que presentan el problema como una secuencia de instantáneas o de grafos que evolucionan en el tiempo.

Algunos de los trabajos que consideran patrones en grafos dinámicos tratan las relaciones mediante rasgos numéricos y la predicción se modela mediante series de tiempo [1], [19], [30], [31], [45]. Otros realizan la predicción de enlace aplicando técnicas de minería de subgrafos frecuentes y reglas de asociación sobre grafos dinámicos [6], [8], [42]. Mención aparte debe hacerse del trabajo de Juszczyszyn *et al.*, los cuales proponen el método *TTM-Predictor* basado en matrices de transición de tríadas o TTM (por sus siglas en inglés: *Triad Transition Matrix*) [32]. Estas matrices contienen las probabilidades de transición en dos instantáneas consecutivas entre los 64 tipos de tríadas, estimando las probabilidades como la cantidad de ocurrencias de estas en cada instantánea.

Este Reporte Técnico centra la atención en los métodos de predicción de enlace mediante patrones dinámicos aplicando técnicas de minería de subgrafos frecuentes y reglas de asociación. Las *Reglas de Evolución* de grafos [6], [8], es la primera aproximación que realiza este tipo de minería para el procesamiento de redes sociales; para ello, realizan la identificación de los patrones teniendo en cuenta que las ocurrencias de las relaciones pueden surgir en tiempos diferentes. Una extensión de las Reglas de Evolución para el procesamiento de patrones dinámicos utilizando estas técnicas es la propuesta de Miyoshi *et al.*, en donde se consideran otros tipos de patrones de tiempo en grafos de evolución plasmado en el método TP-Miner [42]. Otra aproximación a los patrones dinámicos son las *Reglas de Formación de Enlaces*, en donde la predicción de enlace se realiza teniendo en cuenta además la topología de dichos patrones [35].

La predicción de enlace es aplicable a una amplia variedad de áreas, tales como: comercio electrónico, en la Web, redes sociales y en las redes bio-informáticas. En el caso de la Web, puede ser utilizado en tareas de creación automática de enlaces [43], [55] a los que los usuarios puedan estar interesados acceder; ejemplo de estos son los enlaces a libros, películas, juegos, entre otros. En el comercio electrónico, uno de los usos más importantes de la predicción es llevado a cabo en los sistemas de recomendación [10], [39], [49]; sitios tales como Mercado Libre, eBay, Amazon, hacen un gran uso de esta funcionalidad con vistas a aumentar sus ventas. En redes sociales, tales como Facebook, LinkedIn y Twitter, la predicción de enlaces posibilita recomendar amigos, aplicaciones, grupos o comunidades, entre muchos otros [3], [12], [26], [40]. En el caso de las redes bio-informáticas (que representan las relaciones bioquímicas en general) puede ser utilizado, entre otras, para predecir posibles interacciones que puedan llegar a ocurrir entre proteínas [4]. En las aplicaciones relacionadas con la seguridad, la predicción de enlace permite identificar asociaciones ocultas de terroristas y criminales o de potenciales relaciones que puedan establecerse [30], [48], [51]. También se han desarrollado trabajos en los que la predicción de enlace facilita el análisis sobre la evolución de las redes [5], [21], las relaciones entre las topologías de estas y las propiedades que poseen [9], [17], [43]. Además, la predicción de enlace puede ser utilizada en los sistemas de recuperación de información, donde se tiene por objetivo la búsqueda de material de naturaleza no estructurada que satisfaga una información necesaria en grandes colecciones de datos [14], [26], [29]. Esto también puede ser visto como la predicción de relaciones entre palabras y documentos u otros elementos del discurso [40].

Este trabajo se estructura como se describe a continuación. En la próxima sección se abordan los conceptos básicos asociados con el tema que se trata en este reporte y que se utilizan en las siguientes

secciones. Luego, se brinda una breve descripción de los grafos dinámicos definidos para la representación de las redes sociales. Más adelante se aborda en primer lugar las Reglas de Evolución y sus características. A partir de la definición de estas, se da introducción a los Patrones de Tiempo; un enfoque basado en las características principales de las Reglas de Evolución pero que extiende varias propiedades de estas. En la siguiente sección se tratan las Reglas de Formación de Enlaces en redes sociales, las que se sustentan en estructuras diádicas y triádicas. En la última sección se realiza un análisis de los métodos abordados, indicando las principales ventajas, desventajas y posibles extensiones de los algoritmos presentados en este reporte.

2 Conceptos preliminares

A continuación se brindan algunas de las definiciones más importantes de la minería de patrones frecuentes que se utilizan en el presente reporte.

Definición 1. Grafo etiquetado.

Un Grafo Etiquetado es una tripleta $G = (V, E, \lambda)$, donde V es el conjunto de vértices, $E \subseteq V \times V$ es el conjunto de aristas del grafo y $\lambda: V \cup E \mapsto \Sigma$ la función que asigna a los vértices y aristas etiquetas del alfabeto Σ .

Usualmente los grafos etiquetados son utilizados para representar grafos estáticos, donde las etiquetas de las aristas pueden representar proximidad o semejanza entre los vértices, tipos de relación, entre otros. Por su parte el alfabeto Σ es utilizado para delimitar los valores que pueden llegar a tomar las etiquetas de los vértices y aristas, asumiendo por simplicidad que el alfabeto es el mismo para todos los grafos que se definan.

Definición 2. Subgrafo y supergrafo.

Sean $G_1 = (V_1, E_1, \lambda)$ y $G_2 = (V_2, E_2, \lambda)$ dos grafos que comparten el mismo conjunto de etiquetas del alfabeto (Σ) y la misma función etiquetadora (λ). Se dice que G_1 es subgrafo de G_2 si $V_1 \subseteq V_2$, $E_1 \subseteq E_2$. En este caso se utiliza la notación $G_1 \subseteq G_2$ y se dice que G_2 es un supergrafo de G_1 .

Definición 3. Isomorfismo.

Se dice que f es un isomorfismo entre dos grafos etiquetados $G_1 = (V_1, E_1, \lambda_1)$ y $G_2 = (V_2, E_2, \lambda_2)$, si $f: V_1 \mapsto V_2$ es una función biyectiva tal que cumple con las condiciones:

- $\forall v \in V_1, \lambda_1(v) = \lambda_2(f(v))$,
- $\forall (u, v) \in E_1, (f(u), f(v)) \in E_2 \wedge \lambda_1((u, v)) = \lambda_2((f(u), f(v)))$.

Si existe un isomorfismo entre G_1 y G_2 , entonces se dice que G_1 y G_2 son isomorfos.

Definición 4. Sub-isomorfismo.

Se dice que existe un sub-isomorfismo de G_1 a G_2 si G_1 es isomorfo a G_3 y $G_3 \subseteq G_2$. Además, se dice en tal caso de que G_1 es sub-isomorfo a G_2 , denotándose como $G_1 \subseteq_s G_2$.

El problema de determinar el isomorfismo entre dos grafos es NP-duro (NP-hard, en inglés) [22]. Usualmente, en los métodos de minería de grafos frecuentes, las pruebas de isomorfismo se abordan utilizando las formas canónicas de representación de los grafos.

Las formas canónicas para representar los grafos consisten en secuencias de símbolos que identifican de manera única a cada grafo. Estos símbolos describen la topología del grafo y el valor de sus

etiquetas, satisfaciendo sus secuencias ciertos cánones o características especiales, según criterios de orden lexicográfico entre los símbolos [25], [53].

3 Grafos dinámicos

En años recientes se ha visto un incremento en las investigaciones que abordan las redes dinámicas y las distintas aplicaciones que estas poseen. El uso de las redes dinámicas viene dado por la necesidad de representar el constante flujo de información que se genera diariamente en todos los ámbitos de la vida. Dichas redes tienen sus orígenes en el incremento de los sistemas automatizados, el desarrollo de las tecnologías y la multiplicidad y complejidad de los tipos de datos que son necesarios procesar.

Un ejemplo clásico de las redes dinámicas son las redes sociales; estas poseen entre sus características principales el constante flujo de información, la tendencia al crecimiento de relaciones entre sus usuarios, el incremento de servicios agregados, entre muchos otros. Para tratar este tipo de problema, los estudios abordados arrojan una tendencia alta al uso de los grafos dinámicos. Estos tipos de grafo permiten la representación inequívoca de las redes dinámicas; o sea, son capaces de representar de forma exacta el estado de una red en cuanto a las entidades y relaciones existentes en esta, además de los momentos de creación de las relaciones entre dichas entidades.

Los grafos dinámicos, o grafos que evolucionan, parten de la definición formal de grafo etiquetado; en este caso se tiene una secuencia de grafos $G_1 \dots G_T$, donde $G_t = (V_t, E_t, \lambda_t)$ representa un grafo etiquetado cualquiera en un instante de tiempo t . Como $G_1 \dots G_T$ es una secuencia de instantáneas o fotografías (*snapshots*) en el tiempo de una misma red, se puede asegurar entonces que $V_t \subseteq V$ y $E_t \subseteq E$. Una de las formas de representar estos grafos es a través de una instantánea sumaria la cual tiene resumida en sí la secuencia de grafos dinámicos durante cierto periodo de tiempo [38].

Definición 5. Instantánea sumaria.

Dada una secuencia de instantáneas $G_1 \dots G_T$, se dice que $G = (V, E, t, \lambda)$ es una Instantánea Sumaria, si G es un grafo etiquetado, donde $V = \bigcup_{t=1}^T V_t$ y $E = \bigcup_{t=1}^T E_t$, siendo $t: E \mapsto \mathbb{R}$ una función que asigna etiquetas de tiempo a las aristas, verificándose que $t(e) = \min\{j \mid e \in E_j\}$ para cada $e = (u, v)$.

Como puede observarse con la definición dada de la función t , cada arista es etiquetada con el tiempo de su primera aparición. Este tiempo pudiera responder a dos posibles situaciones. La primera de ellas es cuando representa al momento de creación de una arista y a su vez esta se conserva en el tiempo; un ejemplo del uso de este tipo de etiqueta puede ser observada en las redes sociales donde un usuario agrega a un amigo en un instante de tiempo y la tendencia en este tipo de redes es que esta relación se mantenga en el tiempo. Aunque una arista pudiera ser eliminada, se asumirá en lo sucesivo que su ocurrencia es suficientemente rara como para ignorarla.

Otra situación en la que puede asumirse una única etiqueta de tiempo es cuando la arista representa un enlace que surgió y desapareció en ese mismo instante de tiempo o instantánea (ver Fig. 1). Un ejemplo de este comportamiento puede ser visto en las redes telefónicas donde un cliente realiza una llamada a otro cliente en un instante de tiempo dado [8], [12], [35], [50].

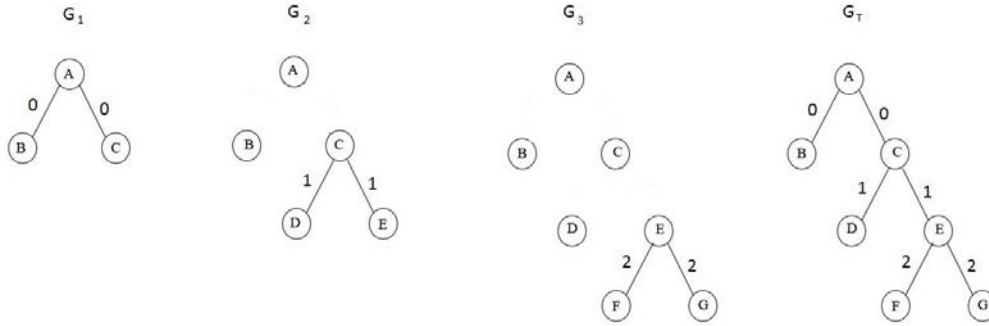


Fig. 1. Ejemplo de secuencia de grafos y de su instantánea sumaria G_T .

La Fig. 1 es un ejemplo de lo antes definido, en esta se observa de izquierda a derecha la secuencia de grafos G_1 , G_2 y G_3 ; dichos grafos representan la evolución de una red durante un intervalo de tiempo dado. Por último, se aprecia la instantánea sumaria G_T , esta tiene resumido dentro de sí el conjunto de vértices y aristas de las tres instantáneas, incluyendo como etiquetas de tiempo el instante (en base 0) en que se produjo la relación.

4 Métodos basados en las reglas de evolución de grafos

Como se comentó con anterioridad, uno de los retos principales que posee la predicción de enlace en las redes sociales es la necesidad de tener en cuenta el dinamismo que estas poseen. Esta problemática ha sido abordada con diferentes aproximaciones.

A continuación se presentan dos métodos de predicción de enlace que consideran lo denominado como Reglas de Evolución de Grafos. Estos métodos consideran sólo la secuencia temporal de aparición de las aristas para predecir el surgimiento de otras en tiempos posteriores. El primero que se presenta es el método GERM, el cual representa el método base, exponiéndose diferentes aspectos generales; o sea, los tipos de patrones temporales, y lo relativo al cálculo del soporte y la confianza. Posteriormente se presenta una extensión del primero basado en los patrones de tiempo en grafos evolutivos y el método asociado (TP-Miner).

4.1 Reglas de evolución de grafos

Las Reglas de Evolución de Grafos (o GER, por sus siglas en inglés: *Graph Evolution Rules*) han sido uno de los primeros enfoques introducidos para la minería de patrones frecuentes basados en el tiempo. Estas poseen como objetivo la predicción de enlace en grafos dinámicos, y fueron introducidas por Berlingerio *et al.* en el 2009 con la finalidad de realizar predicciones de eventos o enlaces en una red social considerando la secuencia temporal de los eventos precedentes [8]. En este trabajo se presenta el método GERM, mostrando mejores resultados que los reportados utilizando otros algoritmos para la predicción de enlace considerando el tiempo [1], [27], [44], [46].

Las GER siguen el esquema **Antecedente** $\rightarrow$ **Consecuente**. El antecedente es conocido también como pre-condición y representa lo que debe garantizarse para que pueda ocurrir el consecuente o pos-condición. Por otro lado, tanto antecedente como consecuente son instantáneas sumarias con similar formato al grafo G_T mostrado en la Fig. 1. En el caso del antecedente, se muestra el patrón identificado hasta un instante de tiempo dado; por su parte, el consecuente muestra el mismo patrón incrementado con nuevas aristas surgidas en un instante de tiempo posterior. De lo anterior se desprende que el

consecuente es un patrón cuyo grafo contiene al antecedente o, lo que es lo mismo, el antecedente es un subgrafo del consecuente. A continuación se formaliza este tipo de regla, en donde las letras B y H se asocian con los patrones antecedente (*Body*) y consecuente (*Head*) de la regla.

Definición 6. Regla de evolución de grafos.

Dadas dos instantáneas sumarias P_B y P_H , se dice que una relación del tipo $P_B \rightarrow P_H$ es una Regla de Evolución de Grafos, si se verifica que:

- $E_B = \{e \in E_H \mid t(e) < \max_{e^* \in E_H} (t(e^*))\}$,
- $V_B = \{v \in V_H \mid \deg(v, E_B) > 0\}$,

donde $\deg(v, E_B)$ es el grado de v con respecto a las aristas de P_B . Se tiene como premisa que P_H es un grafo conexo.

La primera condición define que el conjunto de aristas E_B del antecedente está compuesto por todas las aristas de P_H siempre y cuando las etiquetas de tiempo que posea sean menores que la mayor etiqueta de tiempo de las aristas de P_H . De la segunda condición se desprende que cada vértice de P_B posee un enlace con otro vértice en ese grafo; o sea, ningún vértice estará desconectado del resto del grafo, verificándose particularmente en aquellos que serán enlazados en P_H . La Fig. 2 muestra dos ejemplos de reglas de evolución de grafos. En este caso si se tomara el grafo G como consecuente de la regla, se puede entonces asegurar que Q es el antecedente de la regla de evolución de grafo $Q \rightarrow G$; de igual forma si se tomara a Q como consecuente su antecedente sería el grafo P .

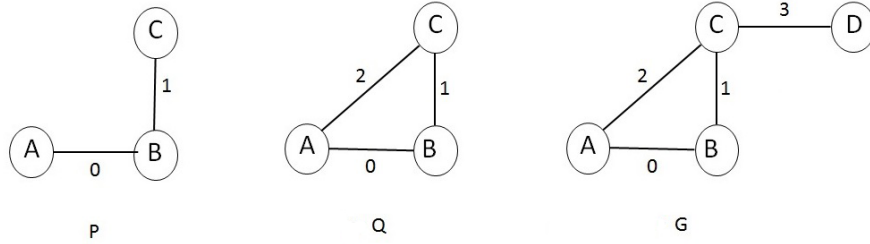


Fig. 2. Dados los grafos P , Q y G , se verifica que $P \rightarrow Q$ y $Q \rightarrow G$ son reglas de evolución de grafos válidas.

4.1.1 Patrones en las reglas de evolución de grafos

En la minería de patrones frecuentes, realizar una correcta identificación de los patrones juega un papel fundamental. El caso de las reglas de evolución de grafos no es la excepción; la principal diferencia con otros es que en este caso se trata de identificar patrones que se encuentran en estructuras que evolucionan en el tiempo y donde las aristas y vértices de dichos patrones poseen un determinado grado de aparición en distintos tiempos y espacios de la red.

Teniendo en cuenta el razonamiento anterior, los autores de GERM definen dos tipos de patrones de tiempo: Patrón de tiempo absoluto y Patrón de tiempo relativo [8].

Definición 7. Patrón de tiempo absoluto.

Sean $G = (V_G, E_G, t_G, \lambda_G)$ y $P = (V_P, E_P, t_P, \lambda_P)$ dos instantáneas sumarias cualesquiera, donde G es el grafo que evoluciona y P un patrón, siendo P conexo. Una ocurrencia de P en G es un Patrón de Tiempo Absoluto si existe una función $\varphi: V_P \mapsto V_G$ de correspondencia de los nodos de P en G y para todas las aristas y vértices se cumple que:

- (i) $\forall (u, v) \in E_P, (\varphi(u), \varphi(v)) \in E_G$,
- (ii) $\forall (u, v) \in E_P, t_G((\varphi(u), \varphi(v))) = t_P((u, v))$,
- (iii) $\forall v \in V_P, \lambda_P(v) = \lambda_G(\varphi(v))$,
- (iv) $\forall (u, v) \in E_P, \lambda_P((u, v)) = \lambda_G((\varphi(u), \varphi(v)))$.

Se puede observar que estas condiciones se cumplen toda vez que se hagan las pruebas de sub-isomorfismo del grafo P con respecto al grafo G y que a su vez se coteje la etiqueta de tiempo de cada arista de P con la etiqueta de tiempo de la arista correspondiente en el grafo G .

Este tipo de patrón ocurre siempre en un mismo instante de tiempo. Ejemplo de este tipo de patrón se puede observar en las redes que modelan noticias. Usualmente, dichas redes realizan publicaciones de noticias nuevas en similares horarios. Por otro lado, en el caso de una red de comunicación puede ser apreciado en los usuarios que sintonizan ciertos programas radiales o televisivos en determinados horarios del día.

Definición 8. Patrón de tiempo relativo.

Sean $G = (V_G, E_G, t_G, \lambda_G)$ y $P = (V_P, E_P, t_P, \lambda_P)$ dos instantáneas sumarias cualesquiera, se dice que P es un Patrón de Tiempo Relativo que ocurre en G si existe un $\Delta \in \mathbb{R}$, la función $\varphi: V_P \mapsto V_G$ de correspondencia de los nodos de P en G , y se cumple que:

- (i) $\forall (u, v) \in E_P, (\varphi(u), \varphi(v)) \in E_G$,
- (ii) $\forall (u, v) \in E_P, t_G((\varphi(u), \varphi(v))) = t_P((u, v)) + \Delta$,
- (iii) $\forall v \in V_P, \lambda_P(v) = \lambda_G(\varphi(v))$,
- (iv) $\forall (u, v) \in E_P, \lambda_P((u, v)) = \lambda_G((\varphi(u), \varphi(v)))$.

La diferencia entre los dos tipos patrones es el Δ en la condición dos. Con esta nueva definición se expresa cómo dos ocurrencias son isomorfas entre si para todas sus etiquetas, menos en las etiquetas de tiempo, las cuales exigen una correspondencia con desplazamiento Δ de todas sus aristas. Un ejemplo de lo anteriormente planteado puede ser visto en la Fig. 3 donde P con Δ igual a 3 es isomorfo con G , siendo G por tanto una ocurrencia de P .

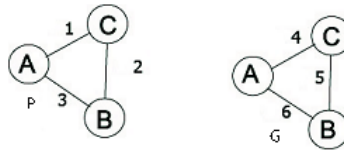


Fig. 3. El grafo P es un patrón de tiempo relativo isomorfo con G con $\Delta = 3$.

Los patrones de tiempo relativo son los utilizados por los autores de GERM para realizar la identificación de las reglas de evolución de grafos, esta elección responde a la necesidad de tener en cuenta la evolución constante de la red, la aparición de las aristas que componen el patrón en tiempos distintos, entre otros.

4.1.2 Cálculo del soporte

El cálculo del soporte se concibe como el conteo de la frecuencia de las ocurrencias de un patrón dentro de una base de datos, en este caso conformada por grafos o instantáneas. Sin embargo, como se indicó en la sección anterior, las ocurrencias de los patrones se buscan en un único grafo masivo, el cual

representa la instantánea sumaria de la red dinámica. El cálculo del soporte se realiza considerando tales características.

Definición 9. Grafo frecuente.

Siendo $Sop(P, G)$ el soporte del grafo P en G , se dice que el grafo P es frecuente en G , si y solo si $Sop(P, G) \geq \delta$, donde δ es el umbral de frecuencia definido.

Calcular el soporte de un patrón en una instantánea no es una tarea trivial. Los autores de varios trabajos han realizado investigaciones que permiten confirmar lo anterior [11], [13], [24], [34]. La condición principal a ser considerada cuando se realiza el cálculo del soporte es la propiedad conocida como Clausura Descendente. La *Clausura Descendente* plantea que: “El soporte de un patrón A es mayor o igual que el soporte de cualquier patrón B que lo contenga; además, si un patrón no es frecuente, entonces ningún super-patrón podrá ser frecuente”. A partir de lo anterior, y aplicándolo a la minería de grafos, es conveniente verificar que todo subgrafo de un grafo frecuente es frecuente, mientras que cualquier supergrafo de un grafo no frecuente tampoco es frecuente [8].

La propiedad de la Clausura Descendente es garantizada en la mayoría de los trabajos de minería de patrones frecuentes reportados hasta el momento [25]. Esto se debe a la posibilidad de realizar efectivas podas en el espacio de búsqueda, a la vez que garantiza que ningún patrón P a medida que vaya creciendo posea una frecuencia mayor que la del patrón que le dio surgimiento. Sin embargo, en la minería de grafos frecuente cuando se consideran todas las ocurrencias (o sub-isomorfismos de tiempo relativo) de un patrón en un único grafo, esta propiedad no siempre se cumple. La Fig. 4 muestra un ejemplo del fenómeno al cual se está haciendo referencia y es que si se conoce que el patrón P_0 es un subgrafo del patrón P , se puede verificar que P ocurre 4 veces en G , mientras que P_0 ocurre sólo una, violándose la Clausura Descendente.

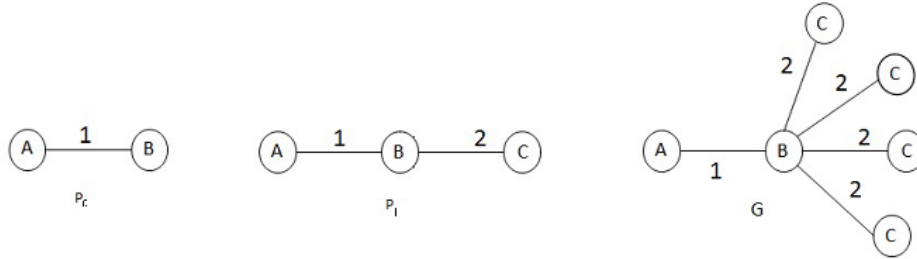


Fig. 4. Ejemplo de violación de la Clausura Descendente de P respecto a P_0 en G .

Tratar de hacer cumplir la Clausura Descendente, trae a discusión otra problemática ampliamente debatida e investigada en la bibliografía abordada: el solapamiento entre ocurrencias de un mismo patrón, la cual es lo que motiva la violación de esa propiedad. El solapamiento ocurre cuando dos subgrafos de un grafo cualquiera (que son ocurrencias de un patrón) tienen solapados entre si vértices y/o aristas. Este fenómeno trae como consecuencia que sea mayor la cantidad de ocurrencias del grafo patrón que la de algunos de sus subgrafos, toda vez que estos se solapen entre sí.

El hecho de que se realice la búsqueda de los patrones sobre una sola instantánea, que se trate de evitar el solapamiento entre las ocurrencias de un mismo patrón y que a su vez se deba de tener en cuenta la Clausura Descendente, trae a relucir la necesidad de buscar estrategias mediante las cuales se pueda definir de la forma más conveniente posible el concepto de *Ocurrencia* de un patrón. Particularmente, se debe garantizar que el soporte del patrón no viole la Clausura Descendente y que a su vez sea lo más semejante posible a la cantidad de ocurrencias reales del patrón en la instantánea sumaria.

Varias son las formas identificadas para realizar el cálculo del soporte. La primera de estas variantes fue propuesta por Kuramochi y Karypis, los que introducen una medida para el cálculo del soporte de un patrón basados en el cálculo del *Máximo Conjunto Independiente*. El *Máximo Conjunto*

Independiente es una definición de la teoría de grafos que maximiza los conjuntos independientes; o sea, los conjuntos no conexos que existan [18].

Definición 10. Conjunto independiente.

Dado un grafo no dirigido $G = (V, E)$, un subconjunto $U \subseteq V$ es un conjunto independiente si en él no existen dos vértices que sean adyacentes.

Basados en la definición anterior, el *Máximo Conjunto Independiente* (MIS, por sus siglas en inglés: *Maximal Independent Set*) es aquel que al agregar un vértice se incluye al menos una adyacencia, dejando de ser conjunto independiente, no existiendo otro de mayor cardinalidad con tales características.

Considerando el concepto de MIS, para calcular el soporte de un patrón en un grafo único, se toman las ocurrencias del patrón en el grafo único como vértices de un nuevo grafo que se construye, existiendo una arista entre dos vértices si las ocurrencias correspondientes se solapan de una forma dada. Con estas consideraciones, la cardinalidad del MIS sobre el grafo construido se toma como el soporte del patrón. El grafo construido de esta forma se denomina como *Grafo de Solapamiento* (en inglés: *Overlap Graph*). Debe observarse que el método MIS es NP-completo, de ahí las limitaciones que se indicarán posteriormente [22], [34].

En la Fig. 5 se muestra un ejemplo del grafo de solapamiento que es construido a través del método MIS para el cálculo del soporte del patrón p . El grafo g de la figura posee tres ocurrencias del patrón; el grafo que se muestra a la derecha es el que se construye una vez que han sido identificadas las ocurrencias de dicho patrón y se obtienen los solapamientos que existen entre ellas. Para este grafo de solapamiento, el tamaño del MIS es igual a dos (ya que ese es el tamaño del mayor conjunto de nodos no adyacentes; o sea, no conectados directamente) y por ende la frecuencia de p en g es dos.

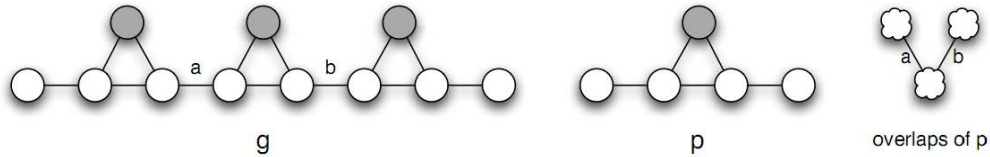


Fig. 5. Grafo de solapamiento del patrón p en g .

La primera estrategia planteada por Bringmann y Nijssen lleva por nombre Solapamiento Simple (*Single Overlap*) y va encaminada a identificar cualquier tipo de solapamiento entre vértices identificado entre dos ocurrencias cualesquiera de una instantánea que se este minando.

Definición 11. Solapamiento simple (Single Overlap).

Existe un solapamiento simple si para las ocurrencias φ y φ' del patrón p se cumple que $\varphi(V_p) \cap \varphi'(V_p) \neq \emptyset$.

En la definición anterior, y en lo que sigue, la expresión $\varphi(V_p)$ se refiere al conjunto de vértices de la ocurrencia φ al corresponderse con cada vértice del patrón P . La frecuencia basada en el solapamiento simple (o sea, que aplica MIS al grafo de solapamiento resultante) se denotará como σ_s .

Puede comprobarse que esta frecuencia satisface la propiedad de la Clausura Descendente. Sin embargo, por aplicarse el procedimiento MIS, su solución es NP-completa. Un ejemplo que ilustra esta definición puede ser visto en la Fig. 6; en este caso existe un solapamiento simple entre las dos ocurrencias del patrón P en uno de los vértices con etiqueta X . Cada vez que se define el solapamiento que se tendrá en cuenta y se identifiquen todas las ocurrencias existentes, se da paso entonces a realizar el cálculo del soporte para el patrón.

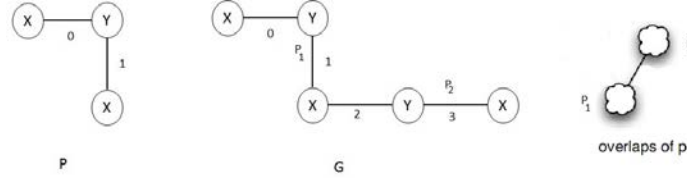


Fig. 6. Ejemplo de grafo de solapamiento de P en G considerando solapamiento simple.

Otra forma de calcular la frecuencia fue introducida por Borgelt *et al.* [23]; en dicho trabajo se propone el Solapamiento Dañino (*Harmful Overlap*), el cual se define de la siguiente forma:

Definición 12. Solapamiento dañino (*Harmful Overlap*).

Existe un solapamiento dañino si para las ocurrencias φ y φ' del patrón P se cumple que: $\exists v \in V_P, \varphi(v), \varphi'(v) \in \varphi(V_P) \cap \varphi'(V_P)$.

La frecuencia basada en el solapamiento dañino se denotará como σ_h . Al igual que el solapamiento simple, esta satisface la propiedad de la Clausura Descendente. El objetivo de este criterio es descartar algunos solapamientos simples manteniendo la satisfacción de esa propiedad. Un ejemplo de esta definición puede ser visto en la Fig. 7, donde se observa el solapamiento existente entre las ocurrencias P_1 y P_2 en uno de los nodos con etiqueta X . Nótese que considerando el solapamiento dañino aplicado a la Fig. 6, el grafo de solapamiento contiene dos nodos P_1 y P_2 no enlazados, por lo que su frecuencia en esa figura sería dos. Al igual que el solapamiento simple, una vez validadas las ocurrencias existentes y el solapamiento considerado, se aplica el método MIS para realizar el cálculo de la frecuencia.

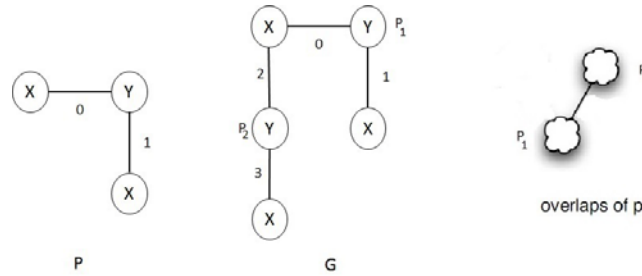


Fig. 7. Ejemplo de grafo de solapamiento de P en G considerando el solapamiento dañino.

Debido a la complejidad NP-Completa que posee el método MIS y los buenos resultados obtenidos durante las pruebas, los autores de GERM proponen otra métrica denominada *Minimum Image Based Support* (que se denotará como σ_m) para realizar el cálculo de la frecuencia y soporte. Dicha métrica permite que el valor que se obtenga cumpla la propiedad de la Clausura Descendente y que además se realice el cálculo disminuyendo el costo computacional. *Minimum Image Based Support* está basado en el número único de vértices en un grafo $G = (V_G, E_G, \lambda_G)$ donde los nodos del patrón $P = (V_P, E_P, \lambda_P)$ son cotejados.

Definición 13. Minimum Image Based Support.

Dada la función $\varphi_i: V_P \mapsto V_G$, por cada ocurrencia φ_i , se define la frecuencia de P en G como:

$$\sigma_m(P, G) = \min_{v \in V_P} |\{\varphi_i(v): \varphi_i \text{ es una ocurrencia de } P \text{ en } G\}|$$

Esta definición puede ser interpretada de la siguiente forma: dado el conjunto de vértices V_P y la lista de ocurrencias del patrón P dentro de un grafo G , se obtiene una lista de conjuntos, uno por cada vértice de V_P , en donde cada conjunto contiene los vértices en G que son cotejados con el vértice de V_P correspondiente entre todas las ocurrencias de P en G ; la frecuencia del patrón será igual a la menor cardinalidad entre todos los conjuntos de la lista. Un ejemplo de los resultados que arroja este método puede ser apreciado en la Fig. 8, donde el patrón posee 3 ocurrencias según esta definición.

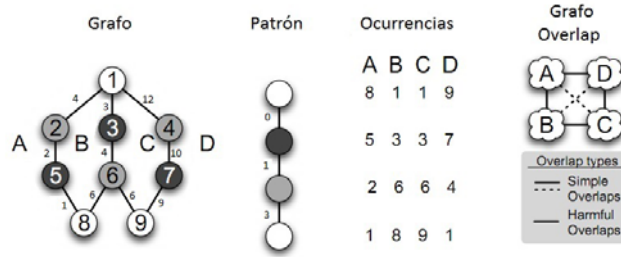


Fig. 8. Ejemplo de grafo con 4 ocurrencias del patrón. Las frecuencias según cada medida son: $\sigma_s = 1 < \sigma_h = 2 < \sigma_m = 3$.

Con respecto a los beneficios que brinda el utilizar esta variante para el cálculo del soporte se encuentra el sensible ahorro de recursos que requiere el método respecto al MIS. La complejidad del algoritmo es polinomial en el peor de los casos y el valor resultante se acerca lo mayor posible a la cantidad real de ocurrencias preservando la propiedad de la Clausura Descendente.

4.1.3 Cálculo de la confianza

En la minería de patrones frecuentes la confianza de una regla de asociación representa una estimación que permite conocer que tan probable es que ocurra el patrón consecuente dado la satisfacción del antecedente de la regla. En el caso de las reglas de evolución, el concepto de confianza y su forma de cálculo es igual al de las reglas de asociación tradicionales [41].

La confianza de una regla $X \rightarrow Y$, expresada como $Conf(X \rightarrow Y)$, se define en función del soporte como:

$$Conf(X \rightarrow Y) = \frac{Sop(X \rightarrow Y)}{Sop(X)} = \frac{Sop(X \cup Y)}{Sop(X)} = \frac{Sop(Y)}{Sop(X)},$$

donde $Sop(X \cup Y)$ se sustituye por $Sop(Y)$ ya que, por definición, el antecedente es un subgrafo del consecuente.

Aunque la expresión de la confianza no cambia, el valor de esta varía en dependencia de la estrategia que se utilice para calcular el soporte. La estrategia validada por los autores del método GERM es considerando como soporte al *Minimum Image Based Support*. Para este método, como se enunció en la Definición 13, los valores que se obtienen de soporte son los más cercanos posibles a la cantidad de ocurrencias reales de los mismos sin violar la propiedad de la Clausura Descendente y además con un costo computacional inferior al del método MIS. Un ejemplo del uso de esta estrategia puede ser

apreciado en la Fig. 9, donde para el grafo X en (A) con esta estrategia la confianza obtenida es igual 1, al igual que el valor obtenido para Y en (B).

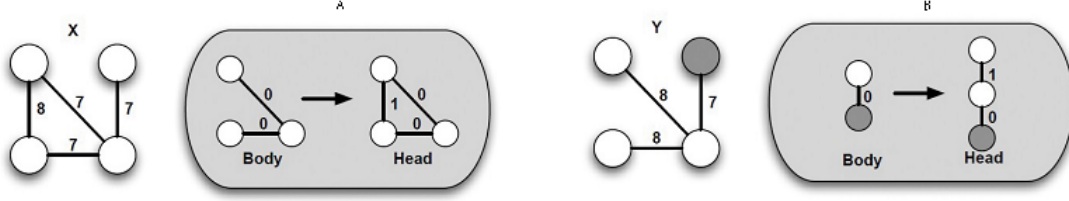


Fig. 9. Ejemplos de reglas de evolución de grafos.

4.1.4 Nuevas métricas de calidad de reglas de evolución

Los autores de GERM propusieron cuatro métricas nuevas. Estas están basadas en el uso del soporte, la confianza y el número de ocurrencias reales del patrón de una regla con el fin de realizar las identificaciones de reglas con una mayor significación para los analistas. Las nuevas métricas parten de la red minada y el conjunto de reglas extraídas a partir del método GERM. Para simplificar las tareas a realizar se utilizan las reglas que tienen una arista p_c de diferencia entre el antecedente y el consecuente, la cual es la arista a predecir [12].

Las nuevas métricas se definen como $score_i(p_c, R)$ y vienen dadas por las siguientes expresiones:

- $score_1(p_c, R) = Conf(R)$;
- $score_2(p_c, R) = Sop(R) * Conf(R)$;
- $score_3(p_c, R) = Conf(R) * count(p_c, R)$;
- $score_4(p_c, R) = Sop(R) * Conf(R) * count(p_c, R)$;

donde R representa la regla que se está procesando, $Sop(R)$ y $Conf(R)$ son el soporte y la confianza respectivamente de dicha regla y $count(p_c, R)$ es la cantidad de ocurrencias reales de la regla dentro de la instantánea. Luego de ser estudiados los resultados de los experimentos, los autores concluyeron que la métrica de mejor comportamiento es $score_3$.

4.1.5 Algoritmo GERM

El algoritmo propuesto por Berlingiero *et al.* fue denominado como GERM (por sus siglas en inglés: *Graph Evolution Rule Miner*) [8]. Para llevar a cabo el diseño e implementación del algoritmo, los autores se basaron en las principales características de gSpan, un algoritmo de minería de patrones frecuentes introducido por Yan *et al.* [53].

Antes de comentar los pasos de GERM es necesario tener en cuenta los siguientes aspectos. En primer lugar una de las principales diferencias entre GERM y gSpan es con respecto a la estructura donde se realiza la búsqueda de los patrones; en el caso del primero se realiza sobre un grafo único o instantánea sumaria, gSpan por su parte realiza este proceso en una colección de grafos. Las colecciones de grafos usualmente contienen grafos de mediana y pequeñas dimensiones, mientras que las instantáneas sumarias suelen ser grafos masivos con vértices densos respecto al grado. Esto trae consigo un aumento considerable en la complejidad del algoritmo. Por tal motivo, en GERM se impone como restricción el número máximo de vértices que poseerá el patrón.

Algoritmo: GERM(G, S, s)

Entrada: G - Instantánea sumaria a minar,
 S - Conjunto de patrones,
 s - Patrón actual

Salida: S - Conjunto de patrones identificados

1: **if** $s \neq \min(s)$ **then return** ;


```

2:  $S \leftarrow S \cup \{s\}$  ;
3: Generar todos los hijos potenciales de  $s$  creciendo con una arista ;
4: Enumerar( $s$ ) ;
5: for all  $c$ ,  $c$  es un hijo de  $s$  do
6:   if  $Sop(c) \geq minSop$  then
7:      $s \leftarrow c$  ;
8:   GERM( $G, S, s$ ) ;

```

Los autores de GERM decidieron utilizar la codificación DFS (por sus siglas en Inglés: *Depth-First Search*) para representar los patrones que se obtienen, la cual es la misma utilizada en gSpan. No obstante, Berlingiero *et al.* extienden la codificación DFS adicionando el tiempo asociado con cada arista del patrón.

Finalmente, durante el proceso de identificación de las ocurrencias de un patrón en GERM, se toma como patrón la ocurrencia que posea como arista inicial la de menor valor de tiempo; esta a su vez se redefine con valor cero, ajustándose en correspondencia los demás tiempos de la ocurrencia. Muestra de lo antes planteado se puede apreciar en la Fig. 10, donde para el patrón P y su lista de ocurrencias se toma la de menor tiempo de aparición; en este caso se podría escoger cualquiera de las dos primeras ocurrencias que se muestra en el listado. En este ejemplo, la frecuencia del patrón, según el método *Minimum Image Based Support* es igual a dos.

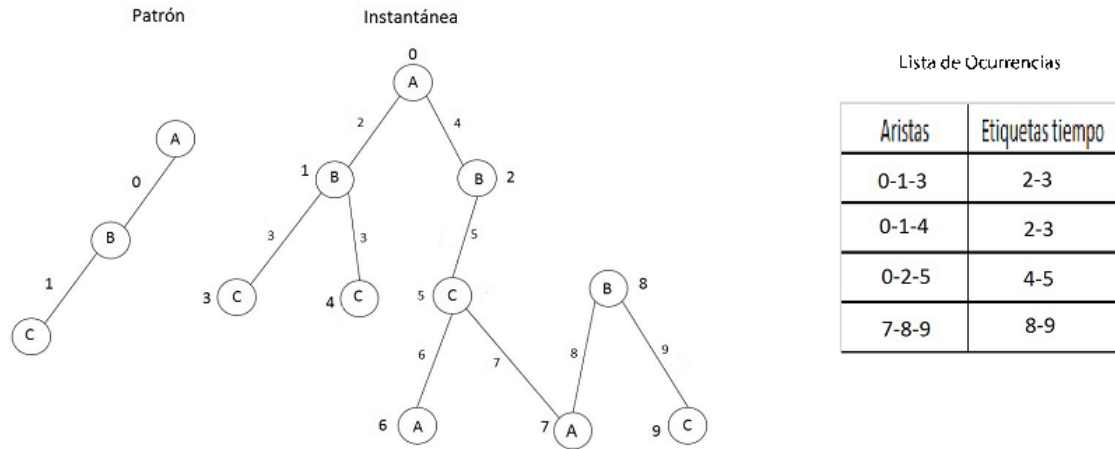


Fig. 10. Lista de ocurrencias del patrón P en G .

El primer paso que se realiza en GERM se encarga de verificar que el código DFS que se está procesando cumpla con ser un código mínimo DFS para el patrón que representa; en caso de no ser así culminaría la llamada recursiva del algoritmo para dicho patrón. Una vez validado el paso anterior y teniendo en cuenta que s es un código DFS mínimo, se agrega a la lista de patrones encontrados. En el tercer paso del algoritmo se buscan todas las posibles aristas a partir de las cuales se puede crecer el patrón s que se está procesando. Dichas aristas son enumeradas y se realiza el crecimiento con cada una de ellas. Una vez realizado dicho crecimiento se da paso a calcular el soporte del nuevo candidato a patrón. En caso de que se compruebe que es frecuente se toma el nuevo subgrafo como patrón y se realiza la llamada recursiva de GERM pasando por parámetro nuevamente, la instantánea sumaria G , el conjunto de patrones S , identificados hasta el momento, y el nuevo patrón a crecer.

4.1.6 Análisis del método GERM

Como se comentó con anterioridad, GERM es el primer trabajo que realiza la minería de patrones frecuentes sobre grafos dinámicos. Este tipo de grafo presenta una estructura que puede resultar muy ventajosa si se tiene en cuenta que:

- Permite representar redes que evolucionan conforme pasa el tiempo.
- Permite conocer para todos los vértices y aristas del grafo el momento en que surgieron.
- Posibilita un sustancial ahorro de recursos, si se tiene en cuenta que la búsqueda de los patrones y la identificación de sus ocurrencias se realiza sobre un solo grafo y el costo de mantenerlo en memoria es mínimo en comparación con las opciones ya conocidas.

Por otro lado, la extensión de la codificación DFS con la adición del tiempo, propuesta por sus autores, permite utilizar este como criterio lexicográfico para el ordenamiento de las aristas. Esto garantiza que los patrones que se vayan identificando tengan ordenadas sus aristas según su orden de aparición, logrando un mejor uso del tiempo y el espacio durante la ejecución del algoritmo.

Los autores de GERM analizaron varias formas para llevar a cabo los cálculos de soporte y confianza, basados principalmente en la idea de que se cumpliera la propiedad de la Clausura Descendente. El método *Minimun Image Based Support* logra los mejores resultados, toda vez que el valor de soporte obtenido es el más cercano posible a la cantidad de ocurrencias reales de los patrones; además, de tener un menor costo computacional.

Amén de lo antes señalado, en GERM se observan algunas insuficiencias o limitaciones. A continuación se presentan las principales que han sido identificadas:

- En el método GERM no se tiene en cuenta las posibles eliminaciones de vértices y/o aristas dentro del grafo. Si se toma en cuenta que el dinamismo es una de las principales características de las redes sociales se podría tomar lo anterior como una deficiencia del algoritmo. No obstante, los autores de GERM sugieren una extensión del método para considerar las eliminaciones de las aristas. En esta extensión se sugiere incluir en las aristas de la instantánea sumaria etiquetas t_I y t_D , correspondientes a los tiempos de surgimiento y extinción del evento asociado con cada arista. Para considerar lo anterior, debe modificarse la condición (ii) de las definiciones 7 y 8; en el caso de la Definición 8 de *Patrón de Tiempo Relativo*, la segunda condición se sustituiría por las dos siguientes:

$$- \quad \forall(u, v) \in E_P, t_{I_G}((\varphi(u), \varphi(v))) = t_{I_P}((u, v)) + \Delta,$$

$$- \quad \forall(u, v) \in E_P, t_{D_G}((\varphi(u), \varphi(v))) = t_{D_P}((u, v)) + \Delta.$$

Tener en cuenta la eliminación de vértices y/o aristas es una temática que no ha sido abordada en los trabajos que realizan predicción de enlaces en grafos dinámicos, sugiriéndose por los autores que las modificaciones anteriores serían suficientes para su tratamiento. No obstante, el tratamiento de las modificaciones en la predicción de enlaces en redes dinámicas tiene matices que no han sido analizadas con la profundidad requerida; esto será tratado en mayores detalles al final del reporte técnico.

- Una de las situaciones que pueden presentarse en las redes sociales son las conexiones múltiples entre los usuarios que la componen. Para tener en cuenta este fenómeno es necesario que la estructura que se utilice permita su representación. El método GERM no permite procesar estos casos, siendo una problemática que debe ser tratada en futuros trabajos.
- Los grafos que poseen multi-atributos en sus vértices y/o aristas permiten modelar una mayor diversidad de problemas. El método GERM no tiene en cuenta esta característica, ya que la minería que realiza solo tiene en cuenta una etiqueta tanto para los vértices como para las aristas, sin considerar el tiempo que se adiciona en la codificación DFS. Tener en cuenta los multi-atributos exigiría una mayor complejidad en los métodos asociados.
- Es conocido que la similitud entre grafos, o partes de estos, es una tarea costosa en tiempo y espacio; además, esta puede realizarse considerando emparejamientos e isomorfismos exactos o inexactos [16]. En GERM se exige que los patrones sean isomorfos exactos en estructuras y etiquetas, permitiendo ciertas inexactitudes en cuanto al tiempo con diferencias constantes en los

patrones de tiempo relativos. Permitir otros tipos de isomorfismos inexactos y otras consideraciones de variabilidad en cuanto al emparejamiento de los grafos es un problema abierto en la predicción de enlace en grafos dinámicos.

- Tener en cuenta los cambios constantes que poseen las redes dinámicas, en cuanto a las etiquetas de vértices y aristas, es una de las problemáticas no tratada en el método GERM. En el caso específico de los grafos dinámicos estos cambios pueden ser representados permitiendo el re-etiquetado de los vértices y aristas del grafo.

4.2 Método TP-Miner

Basados en los enfoques introducidos en GERM, Miyoshi *et al.* consideran los patrones de tiempo en grafos evolutivos propuestos en el método TP-Miner [42]. Este método está ligado intrínsecamente con GERM, ya que los autores sólo realizan cambios en la forma de detectar las ocurrencias, la estructura del grafo a utilizar y el cálculo de la confianza de las reglas, manteniendo todo el procesamiento restante. No obstante, los autores de TP-Miner introducen algunos criterios de cotejo inexacto, pero en un pos procesamiento de los patrones generados, lo que será brevemente comentado al final de esta sección. En los siguientes epígrafes sólo se tratarán los cambios introducidos por los autores respecto al método GERM.

4.2.1 Multiplicidad de aristas

Los autores de este trabajo consideran grafos etiquetados, dinámicos y no dirigidos con multi-aristas para llevar a cabo el proceso de minería. Para lograr lo anterior, las etiquetas de tiempo de las aristas representan, al igual que en GERM, el momento de creación de estas y las etiquetas propias de los nodos y las aristas representan propiedades previamente establecidas. La Fig. 11 muestra un ejemplo de la estructura del grafo presentado.

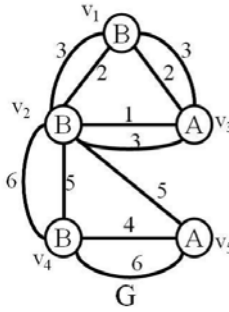


Fig. 11. Ejemplo de grafo dinámico con aristas múltiples utilizado en TP-Miner.

Como se comentó anteriormente, la posibilidad real de que existan interacciones múltiples entre los nodos de una red social es una problemática que no había sido abordada anteriormente en el campo de la predicción de enlaces sobre grafos dinámicos. Estas múltiples interacciones dependen del tipo de red que se esté procesando y de las características propias de sus datos. Introducir esta estructura es una de las principales ventajas que presenta este algoritmo, ya que permite representar y realizar minería sobre redes tales como las de comunicación y correos electrónicos, redes de información, redes telefónicas, entre muchas otras [28], [40].

4.2.2 Cálculo de la confianza

En este trabajo para realizar el cálculo de la confianza no se tuvo en cuenta el valor del soporte del antecedente y del consecuente, tal cuál se realiza en GERM [8] y LFR-Miner [35]. En este caso la

confianza está basada en la cantidad de ocurrencias reales que poseen los patrones dentro de la instantánea, como se indica en la siguiente expresión:

$$Conf_G(P_b \rightarrow P_h) = \frac{|O_{P_b}(P_h)|}{|\Phi_G^{P_b}|},$$

donde $O_{P_b}(P_h)$ denota al conjunto de ocurrencias $o_b \in \Phi_G^{P_b}$ que pueden ser extendibles a ocurrencias $o_h \in \Phi_G^{P_h}$, siendo Φ_G^P el conjunto de ocurrencias del patrón P en la instantánea sumaria G considerando patrones de tiempo relativo.

El cambio está fundamentado en dos aspectos principales; en primer lugar, el valor de la confianza no tiene que satisfacer la propiedad de la Clausura Descendente. Además, considerar la cantidad real de ocurrencias que poseen los patrones permite estimar mejor la probabilidad para la predicción de enlace.

En el método TP-Miner se considera como regla válida aquella cuya confianza es mayor o igual a un umbral mínimo dado.

4.2.3 Análisis del método

Lo primero que debe señalarse sobre TP-Miner es que este presenta las mismas desventajas anotadas para GERM. Respecto a sus ventajas, se observa lo siguiente.

La primera característica de este método que lo distingue es el permitir grafos con multi-aristas, lo que amplía significativamente el campo de aplicaciones. Como se había indicado anteriormente, varios son los problemas que requieren estos tratamientos. Sólo a modo de ejemplo, en los sistemas de recomendación, donde los nodos pueden representar tanto a individuos como a productos, este enfoque permitiría modelar el multi acceso de los clientes a los productos.

La segunda ventaja de este método respecto a GERM es el considerar como confianza valores de ocurrencia en lugar del soporte, dado que este último no coincide con la cantidad real de ocurrencias para permitir que se cumpla la Clausura Descendente. Esto permite identificar reglas más interesantes para los usuarios ya que estiman mejor los valores probabilísticos que usualmente se han asociado con la confianza.

Como fue dicho al inicio, los autores de TP-Miner introdujeron formas de cotejo inexacto pero como pos procesamiento para reducir la cantidad de reglas generadas. Para ello, sobre el universo de los patrones generados, una vez ejecutado el algoritmo básico, ejecutan una serie de pasos para crear un nuevo conjunto de patrones garantizando dos aspectos: (1) la representatividad estructural y (2) la preservación del soporte.

La representatividad estructural lo tratan considerando que dos patrones diferentes se traten como semejantes siempre que difieran en una cantidad máxima de aristas, dado por un umbral límite, pero siendo idénticos en cuanto a las etiquetas de sus vértices. Para la preservación del soporte los autores proponen diversas consideraciones sobre los soportes de los patrones semejantes y de los inducidos por estos. De esta forma, logran comprimir el conjunto de patrones, obteniendo un conjunto que denominan conjunto de *Patrones de Tiempo Representativos*. Las consideraciones asociadas con este concepto y reflejados en el método TP-Miner no serán abordadas en este reporte, pudiendo consultarse el trabajo referido [42].

5 Reglas de formación de enlaces

Las Reglas de Formación de Enlaces (LFR por sus siglas en inglés: *Link Formation Rules*) fueron presentadas a través del método LFR-Miner por Leung *et al.*, con el objetivo de realizar predicciones de enlaces en grafo dinámicos [35]. Este método, aún cuando considera el tiempo en el procesamiento, difiere significativamente del realizado en GERM.

Este trabajo no tiene en cuenta el orden de aparición de las aristas para la identificación de los patrones en los antecedentes de las reglas, aunque sí exige la precedencia en el tiempo de las aristas del antecedente respecto al consecuente. Esto posibilita cierta flexibilidad en los tiempos de surgimiento de los eventos condicionantes. Además, definen estructuras topológicas específicas para sus patrones, basados en las reglas de transitividad relacionadas con las redes sociales; o sea, dos amigos pueden llegar a relacionarse en un futuro en la medida que tengan otros amigos en común.

5.1 Estructura del grafo dinámico

Los grafos dinámicos en los que se aplican las reglas de formación de enlaces son, al igual que en GERM, grafos masivos que resumen en la última instantánea los tiempos de surgimiento de todas las aristas; o sea, consideran lo definido como instantánea sumaria en cuanto al tiempo y su persistencia. Sin embargo, los grafos con los que trabaja LFR-Miner son dirigidos; por ello, se denominarán a estas como *Instantáneas Sumarias Dirigidas*. La Fig. 12 muestra un ejemplo de este tipo de grafo.

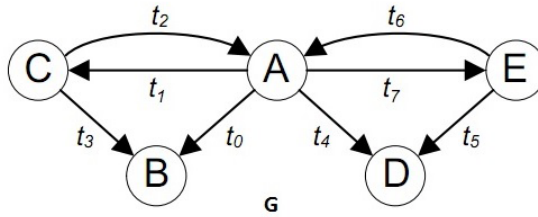


Fig. 12. Ejemplo de grafo dinámico en LFR (Instantánea Sumaria Dirigida).

En esta figura (y en todas las que se presentan) los autores asumen por simplicidad que todas las aristas poseen la misma etiqueta por lo que es omitido en el grafo. No obstante, el algoritmo propuesto considera la posibilidad de tener aristas con diferentes etiquetas.

5.2 Patrones y reglas de formación de enlace

Los *Patrones de Formación de Enlace* (LF, por sus siglas en inglés: *Link Formation*) fueron definidos a partir de la necesidad de que los patrones detectados por el algoritmo no consideren el momento y el orden de creación de las aristas y se concentren solamente en que la arista a predecir ocurra en un tiempo posterior al resto de las aristas del patrón. Además, fue definido que los patrones LF deben estar conectados o débilmente conectados; o sea, que exista al menos un camino entre dos vértices cualesquiera del patrón, omitiendo la dirección de los enlaces. Para lograr esto se basan en la topología de los patrones y restringen la forma y tamaño que podrían llegar a tomar estos, exigiendo estructuras diádicas o triádicas entre los vértices que se enlazan en la predicción. Para ello se plantea que todos los vértices que componen el patrón están conectados con dos nodos centrales, uno denominado focal *s*, también llamado como nodo de partida, y otro denominado final *e*, o nodo terminal (ver Fig. 13).

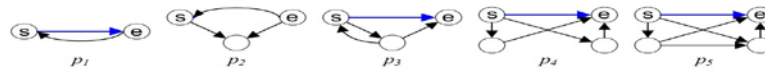


Fig. 13. Ejemplos de patrones LF; en azul la arista a predecir.

Los patrones LF que se minan pueden contener desde simples estructuras diádicas y triádicas (ver patrones *p*₁ y *p*₂ de la Fig. 13) hasta estructuras múltiples diádicas y/o triádicas entre grupos de nodos (ver patrones *p*₃, *p*₄ y *p*₅ de la Fig. 13). A continuación se presenta la definición dada de patrón LF.

Definición 14. Patrón de formación de enlace (patrón LF).

Un Patrón de Formación de Enlace, o Patrón LF, es un grafo etiquetado P , donde V_P contiene (i) el subconjunto $S_P = \{s, e\}$, siendo s y e los nodos de partida y final de P respectivamente; (ii) un conjunto de intermediarios I_P entre s y e , siendo $I_P = V_P \setminus S_P$, pudiendo ser el conjunto vacío ($I_P = \emptyset$). En el conjunto de aristas E_P , cada elemento (u, v) es una arista dirigida de u a v , $u \neq v$, y $(s, e) \in E_P$. Además, se satisfacen las siguientes condiciones:

1. Si $I_P = \emptyset$, entonces $\forall u, v \in S_P, \rho(u, v) = \rho(v, u) = 1$,
2. Si $I_P \neq \emptyset$, entonces $\forall u \in S_P, v \in I_P, \min(\rho(u, v), \rho(v, u)) = 1$.

La expresión $\rho(u, v)$ representa la distancia geodésica de u a v ; o sea, el número de aristas del camino más corto de u a v . Nótese que esa expresión no es simétrica dado que los grafos son dirigidos. Si no existe un camino entre esos nodos, entonces $\rho(u, v) = +\infty$. La primera condición se aplica a patrones diádicos, exigiendo la conexión en ambos sentidos entre el par de nodos; esto se puede observar en el patrón p_1 de la Fig. 13. La segunda condición se aplica a patrones triádicos (simples o múltiples), indicando que todo nodo intermediario tiene que estar conectado directamente en cualquier dirección con ambos nodos centrales s y e . La condición 2 se verifica en los patrones p_2 a p_5 de la Fig. 13.

Para verificar el isomorfismo entre los patrones LF se considera lo conocido (ver Definición 3), exigiendo además que los vértices s y e de ambos patrones coincidan. Un ejemplo de esta especificación del isomorfismo puede apreciarse en la Fig. 13. En esta, el patrón p_1 es sub-isomorfo con p_3 , según la Definición 3; sin embargo, los vértices centrales no coinciden por lo que invalida la condición introducida por los autores. Para expresar el sub-isomorfismo de patrones LF entre p y q se utilizará en lo que sigue la notación: $p \subseteq_{LF} q$.

Las Reglas de Formación de Enlaces, o Reglas LF, permiten realizar la predicción de los enlaces en los cierres diádicos y triádicos.

Definición 15. Regla de formación de enlace (Regla LF).

Una Regla de Formación de Enlace, o Regla LF, sobre un patrón p se expresa como:

$$r(p) = p^a \rightarrow p,$$

donde p^a es el antecedente de $r(p)$, que contiene un conjunto de vértices $V_{p^a} = V_p$ y un conjunto de aristas $E_{p^a} = E_p \setminus \{(s, e)\}$. La arista $(s, e) \in E_p$ es la arista a predecir y es lo que diferencia el consecuente del antecedente.

5.3 Ocurrencia y frecuencia de un patrón LF

Basados en la idea de que las aristas a predecir se forman a partir de un nodo focal y que desde esta arista se identifican los patrones LF, los autores proponen la noción de *Ocurrencia Basada en Ego* (*Ego-Based Occurrence*) y a la vez el cálculo de su frecuencia dada esta definición.

Definición 16. Ocurrencia basada en ego (Ocurrencia ego).

Dado un nodo focal w , conocido como ego, en un grafo G , la Ocurrencia Basada en Ego, Ocurrencia Ego o simplemente ocurrencia, de un patrón p , es un subgrafo $\tilde{p} \subseteq G$ que es isomorfo a p , tal que $\tilde{p}.s = w$, y el tiempo de la arista $(\tilde{p}.s, \tilde{p}.e)$ satisface la siguiente condición:

$$t((\tilde{p}.s, \tilde{p}.e)) > \max_{e \in E_{\tilde{p}} \setminus (\tilde{p}.s, \tilde{p}.e)} t(e).$$

El conjunto de las ocurrencias basadas en ego de p teniendo a w como su nodo de partida se denota como Γ_w^p . Si $|\Gamma_w^p| > 0$, entonces se dice que p ocurre respecto a w y que w soporta a p .

Nótese que el término ‘isomorfo’ de la definición anterior no coincide con el de la Definición 3 ni con el isomorfismo LF ($\subseteq_{LF}$) definido entre patrones, dado que se está cotejando una ocurrencia (o sea, un subgrafo de una instancia sumaria dirigida) con un patrón LF.

Basado en la definición de ocurrencia dada anteriormente se introduce el concepto de frecuencia para patrones LF.

Definición 17. Frecuencia basada en ego (Frecuencia ego).

Dado p , y el conjunto Γ_w^p para cada nodo $w \in V_G$ en G , siendo:

$$\delta(p, w) = \begin{cases} 1 & \text{si } |\Gamma_w^p| > 0, \\ 0 & \text{otro caso;} \end{cases}$$

la Frecuencia Basada en Ego, Frecuencia Ego o simplemente frecuencia, de p en G está dada por:

$$frec(p, G) = \sum_{w \in V} \delta(p, w).$$

En otras palabras, $frec(p, G)$ es el número de nodos que actúan como nodos de partida del patrón p en al menos una ocurrencia en G . El uso de esta medida garantiza la propiedad de la Clausura Descendente, de forma tal que dado los patrones p y q , si se cumple que $p \subseteq_{LF} q$, entonces $frec(p, G) \geq frec(q, G)$.

El soporte de un patrón LF y la confianza de una regla $r(p)$ se obtienen de igual forma que en GERM, una vez definido qué es una ocurrencia y cómo calcular la frecuencia. Debe observarse que p^a no es un patrón LF válido, debido a la ausencia del enlace (s, e) . No obstante, es posible calcular sus ocurrencias basadas en ego y su frecuencia, si se tiene en cuenta que dicho enlace pudiera crearse en el futuro.

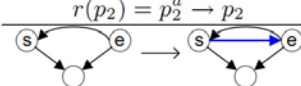
$r(p_2) = p_2^a \rightarrow p_2$	$\text{supp}(r(p_2))$	$\text{conf}(r(p_2))$
	1/5	1/3

Fig. 14. Soporte (supp) y confianza (conf) de la regla LF de p_2 en el grafo G de la Fig. 12.

En la Fig. 14 se muestra el resultado del cálculo del soporte y la confianza para una regla LF generada a partir del patrón p_2 . El grafo G que se muestra en la Fig. 12 es el utilizado en este caso para realizar la minería e identificar la regla. La cantidad de vértices en G es igual a 5, siendo este el valor máximo posible de las ocurrencias de cualquier patrón. El patrón p_2 ocurre respecto a un único ego (el nodo A), en este caso sólo una vez con nodo final E; es por ello que el soporte $\text{Sop}(r(p)) = 1/5$ (supp en la Fig. 14). Por su parte, p_2^a ocurre respecto a tres egos (los nodos A, C y E), por lo que $frec(p_2^a, G) = 3$, siendo entonces la confianza de la regla $\text{Conf}(r(p_2)) = 1/3$.

5.4 Algoritmo LFR-Miner

Debido a las características del grafo considerada por los autores, se hizo necesario utilizar una forma de representación que permitiera tener en cuenta todas las etiquetas del grafo, además de las direcciones de los enlaces; para ello, al igual que en GERM, se utilizó la codificación DFS realizando algunas modificaciones [53].

Para utilizar la codificación DFS con grafos dirigidos se propone modelar cada arista como una 6-tupla: $\langle i, j, l_i, l_{(i,j)}, l_j, d_{(i,j)} \rangle$ donde i y j son los índices de los vértices v_i y v_j respectivamente, l_i y l_j las etiquetas de dichos vértices, $l_{(i,j)}$ es la etiqueta de la arista (i, j) y $d_{(i,j)}$ indica la dirección de esa arista. Se considera $d_{(i,j)} = \rightarrow$ si la arista parte de v_i hacia v_j , en caso contrario sería $d_{(i,j)} = \leftarrow$.

Como es sabido, la codificación DFS define un orden lexicográfico entre las aristas y entre las secuencias de códigos. En la modificación propuesta en este trabajo se extiende lo conocido para la codificación DFS para incluir la dirección de las aristas. Para ello, se define $<_D$ para tratar el orden lexicográfico de los $d_{(i,j)}$, siendo $d_{(i,j)} = \rightarrow$ lexicográficamente menor que $d_{(i,j)} = \leftarrow$.

Un ejemplo de lo antes enunciado es el siguiente: Si se tienen las aristas $\langle 0, 1, 3, 1, 3, \rightarrow \rangle$ y $\langle 0, 1, 3, 1, 3, \leftarrow \rangle$, se puede afirmar que la primera es lexicográficamente menor que la segunda. El código DFS definido por los autores de LFR-Miner siempre toma s como v_0 y e como v_1 , de esta forma se fuerza que la primera arista del código DFS sea (s, e) . En la Fig. 15 se muestra un ejemplo con las condiciones vistas anteriormente; en ella se tiene un patrón p_2 con un código DFS, denotado como $p_2(a)$, y un patrón p_3 con dos posibles códigos DFS, denotados como $p_3(a)$ y $p_3(b)$, siendo $p_3(a)$ lexicográficamente menor que $p_3(b)$ debido a $<_D$.

Edge	$p_2(a)$	$p_3(a)$	$p_3(b)$
1	$\langle 0, 1, *, 1, *, \rightarrow \rangle$	$\langle 0, 1, *, 1, *, \rightarrow \rangle$	$\langle 0, 1, *, 1, *, \rightarrow \rangle$
2	$\langle 1, 0, *, 1, *, \rightarrow \rangle$	$\langle 1, 2, *, 1, *, \leftarrow \rangle$	$\langle 1, 2, *, 1, *, \leftarrow \rangle$
3	$\langle 1, 2, *, 1, *, \rightarrow \rangle$	$\langle 2, 0, *, 1, *, \rightarrow \rangle$	$\langle 2, 0, *, 1, *, \leftarrow \rangle$
4	$\langle 2, 0, *, 1, *, \leftarrow \rangle$	$\langle 2, 0, *, 1, *, \leftarrow \rangle$	$\langle 2, 0, *, 1, *, \rightarrow \rangle$

Fig. 15. Códigos DFS de p_2 y p_3 , siendo * etiquetas vacías.

A continuación se presenta el algoritmo LFR-Miner propuesto por los autores.

Algoritmo: LFR-Miner($G, \text{minS}, \text{minC}, \text{maxV}$)

Entrada: G - Instantánea sumaria dirigida a minar,
 minS - Soporte mínimo,
 minC - Confianza mínima,
 maxV - Máximo de vértices en los patrones.

Salida: R - Conjunto de reglas LF.

```

1:  $\text{minSCnt} = \text{minS} \times |V|$ ;
2:  $R \leftarrow \emptyset$ ;
3:  $S_1 \leftarrow$  Todos los patrones frecuentes de tamaño 1 en  $G$ ;
4: Ordenar  $S_1$  en orden lexicográfico;
5: for each  $p$  en  $S_1$  do
6:    $p.W \leftarrow$  Egos que soportan a  $p$  en  $G$ ;
7:    $\text{MineRules}(p, \text{minSCnt}, \text{minC}, \text{maxV}, R)$ ;
8: return  $R$ ;
    
```

El algoritmo LFR-Miner, para llevar a cabo la minería de patrones frecuentes, al igual que GERM, está basado en el algoritmo gSpan. El primer paso que realiza el algoritmo es la búsqueda de todas las aristas frecuentes en el grafo; una vez finalizado esta búsqueda, la lista es ordenada lexicográficamente y realizado el proceso de crecimiento para cada una de ella. Es a partir de estas aristas que se realiza el crecimiento de cada uno de los patrones que darán lugar a las reglas LF.

Algoritmo: MineRules($p, \text{minSCnt}, \text{minC}, \text{maxV}, R$)

Entrada: p - Patrón a minar,
 minSCnt - Frecuencia mínima,
 minC - Confianza mínima,

$maxV$ - Máximo de vértices en los patrones.

Salida: R - Conjunto de reglas LF.

```

1:  $Q = \text{GrowAndEnumerate}(p, minSCnt, MaxV)$  ;
2: for each  $q \in Q$  do
3:   if  $|q.W| \geq minSCnt$  then
4:     if  $q$  es un patrón LF then
5:       Generar  $r(q)$  ;
6:       if  $Conf(r(q)) \geq minC$  then
7:         Adicionar  $r(q)$  a  $R$  ;
8:    $\text{MineRules}(q, minSCnt, minC, maxV, R)$  ;
```

El algoritmo define el método auxiliar $\text{MineRules}(p, minSCnt, minC, maxV, R)$ para la búsqueda y crecimiento de las aristas frecuentes. Para ello, este método realiza la búsqueda de todas las posibles extensiones y para cada una de estas calcula en primer lugar si posee una frecuencia mayor que la definida inicialmente. Posteriormente, verifica que sea un patrón LF y, en caso de ser afirmativo, se genera la regla LF. A continuación, si la confianza es mayor que el mínimo establecido se adiciona al conjunto de reglas encontradas. El algoritmo culmina retornando la lista de todas las reglas identificadas durante el proceso de minería.

5.5 Análisis del método LFR-Miner

El método tratado en esta sección introduce varios cambios con respecto a GERM y enfoca el reconocimiento de reglas de formación en estructuras locales, además de suavizar el cotejo exacto en cuanto al orden de aparición de las aristas en un patrón. En lo adelante se verán las principales ventajas e insuficiencias identificadas en LFR-Miner.

La estructura de grafo definida en este trabajo logra representar redes dinámicas teniendo en cuenta la dirección desde donde parten los enlaces. Tener en cuenta la dirección permite modelar diversos tipos de redes sociales, como son las de Facebook y Twitter, en donde es común modelar y predecir los seguidores de otros usuarios, o las redes académicas como ResearchGate relacionado con redes de referencias como de seguidores o, incluso, en otros tipos de redes como las de transcripción genética, entre otras.

La consideración de estructuras diádica y triádicas ha sido una de las formas de predicción de enlaces en redes estáticas, teniendo un alto impacto en el tratamiento de las redes sociales [36], [47], [54]. La combinación de estos tratamientos topológicos con redes dinámicas permite ampliar las potencialidades de la predicción de enlace y su campo de aplicación.

No menos importante es la flexibilización que proponen los autores en el tratamiento del orden de surgimiento de las aristas para el antecedente, al ignorar este y concentrarse en sus características topológicas.

Algo que distingue a este trabajo es la propuesta para el cálculo de las ocurrencias. En cierta medida, lo propuesto se asemeja al *minimum image based support*, pero simplificando el cálculo y la complejidad computacional al considerar sólo las ocurrencias de un único nodo, en este caso el nodo focal.

Por último, la codificación DFS y los cambios realizados a la misma, sin lugar a dudas es una ventaja clave cuando se realice en lo adelante minería de patrones frecuentes sobre grafos dinámicos y dirigidos. Tener en cuenta la dirección de los enlaces dependerá del tipo de aplicación y de la red que será procesada.

No obstante, se observan algunas insuficiencias:

- Aunque el cálculo del soporte se basa en una definición de ocurrencia homologable con la de GERM, la confianza, al calcularse sobre la base de las ocurrencias ego y no de la cantidad real de ocurrencias, no representa una medida certera de reglas interesantes.

- Aunque pareciera que el método permite el tratamiento de múltiples aristas, realmente sólo considera los enlaces recíprocos de ciclos entre nodos. La consideración de múltiples aristas es un tema pendiente como extensión de este trabajo.
- Si bien es cierto que la consideración de estructuras diádicas y triádicas son usuales en el procesamiento de redes sociales [36], su extensión a patrones mayores, como los permitidos en GERM, permitiría ampliar sus campos de aplicación.
- Aunque el no considerar el tiempo en los antecedentes flexibiliza a los patrones y reglas que se generan, esto adolece de no considerar las posibles precedencias que pueden ser condicionantes en la predicción de enlaces.
- Este método, al igual que los anteriores, no considera el tratamiento de múltiples atributos en vértices y aristas, así como las relevancias de estos, aspectos comunes a muchas redes como son las de recomendación.
- Los autores de LFR-Miner no abordan la eliminación y, en general, la modificación de las aristas y vértices en el tiempo, ni en sus etiquetas o re-etiquetado.

6 Consideraciones generales

En las secciones anteriores se presentaron los métodos de predicción de enlace que se basan en la generación de reglas de asociación para modelar los cambios de las redes sociales a partir de las secuencias temporales de los grafos; o sea, de grafos dinámicos. Estos métodos son: GERM, TP-Miner y LFR-Miner.

Entre los principales aspectos positivos de estos métodos deben señalarse los siguientes:

- Logran representar de forma compacta en una instantánea resumida la secuencia de grafos a minar, logrando optimizaciones de espacio en los algoritmos asociados.
- Proponen extensiones a la codificación DFS para adicionar el tiempo y la dirección de las aristas (en el caso de considerarse grafos dirigidos).
- Proponen como medidas del soporte al *Minimum Image Based Support*, en los dos primeros, y la frecuencia basada en ego, en LFR-Miner. Estas medidas, al parecer, son óptimas respecto al número real de ocurrencias, preservando la propiedad de Clausura Descendente. Estas dos medidas fueron construidas sobre la base de la cantidad de nodos diferentes en que ocurren los patrones; sin embargo, aunque homologables, estas difieren en su concepción. Sería interesante realizar un análisis teórico de sus diferencias y semejanzas, y determinar cuál logra valores más cercanos a la cantidad real de ocurrencias.
- El cálculo de la confianza es decisivo para lograr reglas significativas para los usuarios. En este sentido, la propuesta de TP-Miner permite obtener regla que estiman mejor las probabilidades de predicción, por lo que se propone aplicar esta métrica en cualquier extensión que se haga.
- En el método TP-Miner se extiende la predicción a grafos con multi-aristas, ampliando el campo de aplicación de estas técnicas. Se propone evaluar la conveniencia de extender el método LFR-Miner a patrones con multi-aristas.
- Al menos un trabajo (LFR-Miner) permite tratar la dirección de las aristas, aunque sobre estructuras locales (triadas). Se propone extender los otros métodos a grafos dirigidos.
- Ese mismo trabajo acepta una forma de cotejo inexacto al flexibilizar el orden de aparición de las aristas en el antecedente, aunque de forma extrema; o sea, obviándolo totalmente. En este sentido, se considera perfectible este aspecto, como se expondrá en lo que sigue.

Mientras tanto, se pueden identificar una serie de insuficiencias, limitaciones o posibles perfeccionamientos:

- En ninguno de los métodos se presenta un tratamiento completo al problema de las posibles eliminaciones y modificaciones de vértices y/o aristas. Sólo asociado con el método GERM se proponen posibles extensiones considerando en las aristas un tiempo de surgimiento y otro de

eliminación. Sin embargo, no todas las aristas tienen que tener ambos tiempos. Tampoco es preciso que la diferencia del tiempo en los patrones de tiempo relativo sea la misma para la inserción y la eliminación de la misma arista. Otro aspecto a considerar es que un mismo tipo de enlace puede surgir y desaparecer secuencialmente en diferentes momentos, lo cual no es modelado en lo propuesto, aunque pudiera evaluarse su tratamiento con multi-aristas.

- En ninguno de los métodos expuestos se considera el re-etiquetado de las aristas y/o vértices, limitando los problemas de redes sociales con tales características.
- La posibilidad de considerar multi-aristas junto a su dirección es algo que debe valorarse en las extensiones a los métodos presentados.
- En los métodos propuestos no se trata en toda su complejidad la posibilidad del cotejo inexacto. El método TP-Miner permite algunas posibilidades en un pos procesamiento, pero no en la fase inicial de generación de los patrones. Mientras, el método LFR-Miner permite no considerar el orden temporal en el antecedente; sin embargo, en muchas ocasiones el orden de los eventos refuerza las probabilidades de predicción. No obstante, la diferencia de tiempo relativo no necesariamente debe seguirse de forma constante para todas las aristas de un patrón, ni requerirse un orden estricto para todas las aristas. El cotejo inexacto de etiquetas, topología y, sobre todo, el tiempo es una necesidad a investigar con mayor profundidad en un futuro.
- Ninguno de los métodos tratan el problema de los multi-atributos en los vértices y/o aristas, aspecto este que debe considerarse en las extensiones que se hagan. En este punto debe tenerse en cuenta que se han propuesto métodos, como los que predicen enlaces basados en series de tiempo, que trabajan con enlaces numéricos; sin embargo, usualmente los atributos de las aristas y/o nodos pudieran ser de diversa naturaleza, exigiéndose considerar la generación de patrones con datos mezclados e incompletos, lo que no ha sido tratado hasta el presente.
- Considerando que existen otros métodos de predicción de enlace, particularmente sobre grafos estáticos (como los métodos topológicos), que consideran características convenientes en el tratamiento de las redes sociales, se propone extender esos métodos teniendo en cuenta lo anterior. En este aspecto se distingue el método LFR-Miner que trabaja sobre estructuras locales particulares (estructuras diádicas y triádicas); sin embargo, existen otras estructuras sobre subgrafos más extensos que deben ser consideradas. Además, otras características de las redes tratadas por otros métodos (la comunidad de vecinos, sus densidades, las distancias entre nodos, los roles que estos juegan, etc.) pueden resultar apropiadas combinarlas con los métodos aquí expuestos.
- El costo computacional de estos métodos en cuanto al tiempo es alto, dependiendo fundamentalmente de los costos de los métodos de generación de subgrafos frecuentes que se apliquen. Definir algoritmos que logren disminuir dichos tiempos es un reto en este campo.

Conclusiones

La predicción de enlace es una de las líneas que ha experimentado un significativo y creciente desarrollo en los últimos años. Esto se sustenta en la necesidad de analizar la evolución de los enlaces de las redes sociales, en cuanto al surgimiento, desaparición o modificación de las relaciones sociales, previendo comportamientos futuros, así como permitiendo evaluaciones sobre comportamientos pasados o actuales.

En este reporte se analizó el estado del arte sobre la predicción de enlace en grafos dinámicos considerando la minería de patrones frecuentes y reglas de asociación. Para ello, se analizaron los trabajos relacionados con los tres métodos que, hasta nuestro conocimiento, han sido propuestos: GERM, TP-Miner y LFR-Miner. En la sección “Consideraciones generales” se presentaron diferentes aspectos que se consideran las principales fortalezas de estos métodos, así como las insuficiencias observadas.

Como puede observarse, se propone trabajar en diversas direcciones, con diferentes complejidades. Entre estas resaltan las extensiones de los métodos propuestos en cuanto a la dinámica de los enlaces y nodos (creación, eliminación, modificación y re-etiquetado). Sin embargo, aún más importante es la inclusión de otros criterios (topológicos, de roles, etc.) en los métodos que se propongan.

Tema aparte, por su complejidad, merece el tratamiento de los diferentes cotejos inexactos que deben tenerse en cuenta para la predicción de enlace en grafos dinámicos.

Otro tema de gran complejidad, pero que se percibe como necesario su abordaje, es el tratamiento de multi-atributos con la posibilidad de considerar datos mezclados e incompletos.

Dada las diversas direcciones de trabajo y sus complejidades, los trabajos a realizar deben estar matizados por las características de las redes sociales que en la actualidad ocupan y preocupan a los especialistas y analistas de las redes sociales.

Como se ha podido apreciar, en este reporte técnico se ha presentado una visión general de una parte de un problema muy vasto, pero de vital importancia para la sociedad actual. Su objetivo ha sido, y se espera se haya logrado, que este sirva de punto de partida para futuras investigaciones básicas y aplicadas sobre la predicción de enlace en las redes sociales.

Referencias bibliográficas

1. Acar, E., Dunlavy, D. M., Kolda, T. G.: Link prediction on evolving data using matrix and tensor factorizations. In: Data Mining Workshops, ICDMW'09, IEEE International Conference on, pp. 262–269 (2009).
2. Adamic, L. A., Adar, E.: Friends and neighbors on the web. *Social networks*, vol. 25, no. 3 pp. 211–230 (2003).
3. Aggarwal, C. C.: An introduction to social network data analytics. *Social Network Data Analytics*, pp. 1–15 (2011).
4. Airoldi, E. M., Blei, D. M., Xing, E., Fienberg, S.: Mixed membership stochastic block models for relational data, with applications to protein-protein interactions. In: Proceedings of International Biometric Society-ENAR Annual Meetings, vol. 5 (2006).
5. Albert, R., Barabási, A. L.: Statistical mechanics of complex networks. *Reviews of modern physics*, vol. 74, no. 1 (2002).
6. Apostolico, A., Erdős, P. L., Györi, E., Lipták, Z., Pizzi, C.: Efficient algorithms for the periodic subgraphs mining problem. *Journal of Discrete Algorithms*, vol. 17, pp. 24–30 (2012).
7. Barabási, A. L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., Vicsek, T.: Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3), pp. 590–614 (2002).
8. Berlingerio, M., Bonchi, F., Bringmann, B., Aristides, G.: Mining graph evolution rules. *Machine learning and knowledge discovery in databases*, Springer, pp. 115–130 (2009).
9. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D-U.: Complex networks: Structure and dynamics. *Physics reports*, vol. 424, no. 4, pp. 175–308 (2006).
10. Breiman, L., Spector, P.: Submodel selection and evaluation in regression. The X-random case. *International statistical review/revue internationale de Statistique*, JSTOR, pp. 291–319 (1992).
11. Bringmann, B., Nijssen, S.: What is frequent in a single graph?. *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, pp. 858–863 (2008).
12. Bringmann, B., Berlingerio, M., Bonchi, F., Gionis, A.: Learning and predicting the evolution of social networks. *Intelligent Systems*, IEEE, 25(4), pp. 26–35 (2010).
13. Calders, T., Ramon, J., Van Dyck, D.: Anti-monotonic overlap-graph support measures. In: Data Mining 2008, ICDM'08, Eighth IEEE International Conference on, pp. 73–82 (2008).
14. Chowdhury, G.: Introduction to modern information retrieval. Facet publishing (2010).
15. Chung, F., Zhao, W.: PageRank and random walks on graphs. *Fete of combinatorics and computer science*, Springer Berlin Heidelberg, pp. 43–62 (2010).
16. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03), pp. 265–298 (2004).

17. Costa, L. D. F., Rodrigues, F. A., Travieso, G., Villas Boas, P. R.: Characterization of complex networks: A survey of measurements, *Advances in Physics*, 56(1), pp. 167–242 (2007).
18. Dharwadker, A.: The independent set algorithm. Institute of Mathematics (2006).
19. Dhote, Y., Mishra, N., Sharma, S.: Survey and analysis of temporal link prediction in online social networks. In: *Advances in Computing, Communications and Informatics (ICACCI)*, International Conference on, IEEE, Pp. 1178–1183 (2013).
20. Doppa, J. R., Yu, J., Tadepalli, P., Getoor, L. (2009). Chance-constrained programs for link prediction. In: *NIPS Workshop on Analyzing Networks and Learning with Graphs* (2009).
21. Dorogovtsev, S. N., Mendes, J. F.: Evolution of networks. *Advances in physics*, 51(4), pp. 1079–1187 (2002).
22. Eppstein, D.: Subgraph isomorphism in planar graphs and related problems. In: *Proceedings of the sixth annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, vol. 95, pp. 632–640 (1995).
23. Fiedler, M., Borgelt, C.: Support Computation for Mining Frequent Subgraphs in a Single Graph. In: *MLG* (2007).
24. Fiedler, M., Borgelt, C.: Subgraph support in a single large graph. In: *Data Mining Workshops, ICDM, Seventh IEEE International Conference on*, pp. 399–404 (2007).
25. Gago-Alonso, A., Carrasco-Ochoa, J. A., Medina-Pagola, J. E.: Minería de subgrafos conexos frecuentes en colecciones de grafos etiquetados. Technical Report CCC-09-003, National Institute of Astrophysics, Optics and Electronics (INAOE), Puebla, Mexico (2009).
26. Getoor, L., Diehl, C. P.: Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2), pp. 3–12 (2005).
27. Al Hasan, M., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: *SDM'06 Workshop on Link Analysis, Counter-terrorism and Security* (2006).
28. Al Hasan, M., Zaki, M. J.: A survey of link prediction in social networks. In: *Social network data analytics*, Springer US, pp. 243–275 (2011).
29. Huang, Z., Li, X., Chen, H.: Link prediction approach to collaborative filtering. In: *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pp. 141–142 (2005).
30. Huang, Z., Lin, D. K. J.: The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing*, 21(2), pp. 286–303 (2009).
31. Ibrahim, N.M.A., Chen, L.: Link prediction in dynamic social networks by integrating different types of information. *Applied Intelligence*, vol. 42, no. 4, pp. 738–750 (2015).
32. Juszczyszyn, K., Budka, M., Musial, K.: The dynamic structural patterns of social networks based on triad transitions. In: *Advances in Social Networks Analysis and Mining (ASONAM)*, International Conference on, IEEE Computer Society, pp. 581–586 (2011).
33. Katz, L.: A new status index derived from sociometric analysis. *Psychometrika*, 18(1), pp. 39–43 (1953).
34. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. *Knowledge and Data Engineering, IEEE Transactions on*, 16(9), pp. 1038–1051 (2004).
35. Leung, C. W. K., Lim, E. P., Lo, D., Weng, J. (2010, October). Mining interesting link formation rules in social networks. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 209–218 (2010).
36. Liaghat, Z., Rasekh, A. H., Mahdavi, A.: Application of data mining methods for link prediction in social networks. *Social Network Analysis and Mining*, 3(2), pp. 143–150 (2013).
37. Li, X., & Chen, H. (2009, June). Recommendation as link prediction: a graph kernel-based machine learning approach. In: *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pp. 213–216 (2009).
38. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7), pp. 1019–1031 (2007).
39. Lin, D.: An information-theoretic definition of similarity. In: *ICML*, vol. 98, pp. pp. 296–304 (1998).
40. Lü, L., Zhou, T.: Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6), pp. 1150–1170 (2011).
41. Medina-Pagola, J. E., Hernández-Palancar, J., Hernández-León, R., Pérez-Suárez, A., Hechavarría-Díaz, A., González-Gazapo, R.: Generación de Conjuntos de Ítems y Reglas de Asociación. RT_002 CENATAV, La Habana, Cuba (2007).
42. Miyoshi, Y., Ozaki, T., Ohkawa, T.: Mining interesting patterns and rules in a time-evolving graph. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1 (2011).
43. Newman, M. E.: The structure and function of complex networks. *SIAM review*, 45(2), pp. 167–256 (2003).

44. Taskar, B., Wong, M. F., Abbeel, P., Koller, D.: Link prediction in relational data. In: Advances in neural information processing systems (2003).
45. Sarkar, P., Chakrabarti, D., Jordan, M.: Nonparametric link prediction in dynamic networks. arXiv preprint arXiv:1206.6394, (2012).
46. Sarukkai, R. R.: Link prediction and path analysis using Markov chains. *Computer Networks*, 33(1), pp. 377–386 (2000).
47. Schall, D.: Link prediction in directed social networks. *Social Network Analysis and Mining*, vol. 4, no 1, pp. 1–14 (2014).
48. Shaikh, M. A., Wang, J., Yang, Z., & Song, Y. (2007). Graph structural mining in terrorist networks. In :Advanced data mining and applications, Springer Berlin Heidelberg, pp. 570–577 (2007).
49. Sun, D., Zhou, T., Liu, J. G., Liu, R. R., Jia, C. X., Wang, B. H.: Information filtering based on transferring similarity. *Physical Review E*, 80(1), (2009).
50. Sun, Y., Han, J., Aggarwal, C. C., Chawla, N. V.: When will it happen?: relationship prediction in heterogeneous information networks. In: Proceedings of the fifth ACM international conference on Web search and data mining, pp. 663–672 (2012).
51. Wang, J. H., Lin, C. L.: An association model for implicit crime link analysis. In: *Intelligence and Security Informatics*, Springer Berlin Heidelberg, pp. 15–21 (2010).
52. Wang, P., Xu, B., Wu, Y., Zhou, X.: Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38 (2015).
53. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: Proceedings of 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi, Japan, pp. 721–724 (2002).
54. Zhang, J., Wang, C., Wang, J., Yu, P. S.: LaFT-tree: perceiving the expansion trace of one's circle of friends in online social networks. In: Proceedings of the sixth ACM international conference on Web search and data mining, pp. 597–606 (2013).
55. Zhou, T., Ren, J., Medo, M., Zhang, Y. C.: Bipartite network projection and personal recommendation. *Physical Review E*, 76(4) (2007).

RT_031, septiembre 2015

Aprobado por el Consejo Científico CENATAV

Derechos Reservados © CENATAV 2015

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

