RNPS No. 2143
ISSN 2072-6260
Versión Digital

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

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

La minería de subgrafos frecuentes aproximados para la clasificación de imágenes: estado del arte

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

RT_027 **junio 2014**

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

RNPS No. 2143
ISSN 2072-6260
Versión Digital

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

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

La minería de subgrafos frecuentes aproximados para la clasificación de imágenes: estado del arte

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

RT_027 **junio 2014**

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

Tabla de contenido

Lista de Algoritmos

La minería de subgrafos frecuentes aproximados para la clasificación de imágenes: estado del arte

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

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

RT_027, Serie Gris, CENATAV
Aceptado: 16 de Junio de 2014

Resumen. La minería de subgrafos frecuentes aproximados (MSFA) se ha convertido en una tarea importante que puede ser aplicada en varios dominios de la ciencia. Esto se debe a que en este tipo de minería se permiten distorsiones en los datos con utilidad para algunas tareas concretas.
Mediante un estudio de la MSFA realizado en este trabajo, específicamente de los métodos que permiten aproximaciones semánticas entre etiquetas manteniendo la topología de los grafos, se pudo observar la utilidad del tratamiento de este tipo de distorsiones. En este trabajo se presenta una comparación y un resumen de los resultados obtenidos al aplicar este tipo de minería para la clasificación de imágenes. Por otro lado, algunos autores han enfocado sus esfuerzos en mejorar la eficacia y la eficiencia de estas técnicas obteniendo buenos resultados en clasificación de imágenes. Dichos resultados son resumidos en este trabajo también como parte del estudio realizado.
Existen áreas donde la MSFA no ha sido aplicada aún y pudieran obtenerse resultados relevantes utilizando estas técnicas. Varias de estas áreas son presentadas en este trabajo como fuertes candidatas para la aplicación de la MSFA.

Palabras clave: minería de grafos, subgrafos frecuentes, subgrafos frecuentes aproximados, minería de subgrafos frecuentes aproximados.

Abstract. Frequent approximate subgraph mining has become an important task which could be applied in several domains of the science. This is because in this kind of mining, useful data distortions for some concrete tasks are allowed.
In this work, a study of the frequent approximate subgraph mining for image classification is performed, specifically the methods which treat semantic approximations between labels keeping the graph topologies. Through this study, the usefulness of treating this type of distortions is observed. A comparison and a summary of the results obtained using this kind of mining for image classification is presented in this work. On the other hand, several authors have focused to improving the effectiveness and efficiency of this technique where relevant results are obtained over image classification tasks. These results are summarized too as part of the study performed in this work.
There are areas where frequent approximate subgraph mining has not been yet applied and relevant results could be obtained using these techniques. Several of these areas are discussed in this work as strong candidates to applying frequent approximate subgraph mining.

Keywords: graph mining, frequent subgraphs, frequent approximate subgraphs, frequent approximate subgraph mining.

1. Introducción

La minería de subgrafos frecuentes es una técnica de minería de datos con gran aplicabilidad en diversas áreas de la ciencia donde los datos sean modelados en forma de grafos [1,2,3,4,5,6,7,8,9,10]. Mediante el cálculo de los subgrafos frecuentes se han logrado buenos resultados en tareas de clasificación de imágenes [6,11,12], clasificación de estructuras moleculares y bioquímicas [2,3,4,5,8,10], así como en el análisis de vínculos y redes sociales [7,9]. Sin embargo, fueron detectados problemas en la práctica donde se limita el uso de estos algoritmos, o donde los patrones calculados en este tipo de minería no son de utilidad [6,13]. Esto se debe a que en los problemas prácticos no existen objetos exactamente iguales y las distorsiones entre los datos de los objetos no son tratadas en este tipo de minería. Por tal motivo surge la necesidad de evaluar la semejanza entre grafos permitiendo diferencias en los datos. Desde entonces, se han presentado diferentes funciones y heurísticas para la evaluación de la semejanza entre grafos [14].

Teniendo en cuenta este hecho, se comenzaron a desarrollar algoritmos para la minería de subgrafos frecuentes aproximados (MSFA) con el objetivo de permitir aproximaciones entre objetos en el proceso de la minería. Al tratar distorsiones en los datos, los patrones calculados por estos algoritmos brindan información más cercana a la realidad. Por este motivo, se han obtenido buenos resultados en diferentes dominios de la ciencia, tales como: clasificación de imágenes [15,16,17,18,19], análisis de redes genéticas y estructuras bioquímicas [20,21,22,23,24], análisis de circuitos, citas, redes sociales y vínculos [6,25]. Sin embargo, solo *VEAM* [15] y *APGM* [21] tratan variaciones semánticas entre las etiquetas manteniendo la topología de los grafos en colecciones de grafos. Estas variaciones entre etiquetas se calculan dada la especificación de cuáles vértices, aristas o etiquetas pueden reemplazar a otras. Esto es útil en muchas aplicaciones prácticas ya que no siempre un vértice, arista o etiqueta puede ser reemplazada por cualquier otra sino que se debe tener en cuenta la semántica de esta sustitución.

En este trabajo se realiza un estudio de la MSFA, específicamente del enfoque utilizado por VEAM ya que es el único que permite distorsiones semánticas entre etiquetas de vértices y aristas, manteniendo la topología de los grafos, en el proceso de minería en colecciones de grafos. Mediante este estudio se muestra un análisis de los resultados y aportes alcanzados con este enfoque. Finalmente, se mencionan algunas de las posibles aplicaciones donde se pudieran obtener resultados relevantes mediante el uso de la MSFA.

Por otro lado, varios trabajos se han enfocado en aumentar la eficiencia y la eficacia de la MSFA. Se han propuesto podas con el objetivo de disminuir el espacio de búsqueda de las etiquetas de los grafos mediante procesamientos iniciales del proceso de minería [21,26,27]. Mediante estas podas se logra una considerable reducción de pruebas de subisomorfismo y de formas canónicas, alcanzando ahorro en tiempo de procesamiento. Otros trabajos se enfocan en la selección de patrones representativos para disminuir la dimensionalidad del conjunto de patrones a utilizar en la clasificación [19]. Estos trabajos se basan en algunas propiedades de los patrones emergentes para el cálculo de un conjunto de patrones más útiles para la clasificación alcanzando buenos resultados en esta tarea. Estos resultados se presentan como parte del estudio realizado en este trabajo.

Este trabajo está organizado de la siguiente manera. En la sección 2 se presentan algunos conceptos básicos y se define el problema de la MSFA. En la sección 3 se presenta la descripción de los trabajos relacionados, así como de un método aproximado y se presenta el diseño de un algoritmo para la MSFA. Los resultados alcanzados sobre varias colecciones de grafos utilizando la MSFA se detallan en la sección 4. Luego, en la sección 5 se presentan las posibles aplicaciones

del proceso de MSFA en diferentes dominios de la ciencia. Finalmente, las conclusiones de este trabajo y algunas ideas de trabajo futuros son expuestas en la sección 6.

2. Marco teórico

En esta sección se presentan los conceptos básicos necesarios para definir el problema de la minería de subgrafos frecuentes aproximados (MSFA), se dan conceptos de clasificación, así como una breve presentación de algunos criterios de selección necesarios para entender el resto del documento. También se describen propiedades presentes en algunos algoritmos del estado del arte.

2.1. Conceptos básicos usados de la teoría de grafos

En este trabajo se trata el procesamiento de colecciones de grafos etiquetados, simples y no dirigidos utilizando como atributos los patrones calculados mediante un algoritmo para la MSFA. En adelante, cuando se hable de grafos se suponen este tipo de grafos, en otro caso se especificará.

Definición 1 (Grafo etiquetado). *Un* grafo etiquetado *es una 5-tupla,* $G = (V, E, L_V, L_E, I, J)$*, donde:*

- V *es un conjunto cuyos elementos son conocidos como* vértices
- $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ *es un conjunto cuyos elementos son conocidos como* aristas *(la arista* $\{u, v\}$ *conecta los vértices* u *y* v*)*
- L_V *es el conjunto de etiquetas para los vértices*
- L_E *es el conjunto de etiquetas para las aristas*
- $I : V \to L_V$ *es una función etiquetadora encargada de asignar etiquetas a los vértices*
- $J : E \to L_E$ *es una función etiquetadora encargada de asignar etiquetas a las aristas*

Definición 2 (Subgrafo y supergrafo). *Sean* $G_1 = (V_1, E_1, L_{V_1}, L_{E_1}, I_1, J_1)$ *y* $G_2 = (V_2, E_2, L_{V_2}, L_{E_2}, I_2, J_2)$ *dos grafos etiquetados, se dice que* G_1 *es un subgrafo de* G_2 *si* $V_1 \subseteq V_2$*,* $E_1 \subseteq E_2$*,* $\forall u \in V_1, I_1(u) = I_2(u)$*, y* $\forall e \in E_1, J_1(e) = J_2(e)$*. En este caso, se utiliza la notación* $G_1 \subseteq G_2$ *y además se dice que* G_2 *es un supergrafo de* G_1*.*

Definición 3 (Isomorfismo). *Dados dos grafos etiquetados* G_1 *y* G_2*, se dice que* f *es un isomorfismo entre esos grafos si* $f : V_1 \to V_2$ *es una función biyectiva, donde:*

- $\forall u \in V_1 : f(u) \in V_2 \wedge I_1(u) = I_2(f(u))$
- $\forall \{u, v\} \in E_1 : \{f(u), f(v)\} \in E_2 \wedge J_1(\{u, v\}) = J_2(\{f(u), f(v)\})$

Si existe un isomorfismo entre G_1 *y* G_2*, se dice que* G_1 *y* G_2 *son isomorfos. Si* G_1 *es isomorfo a* G_3 *y* $G_3 \subseteq G_2$*, entonces se dice que existe un* sub-isomorfismo *entre* G_1 *y* G_2*, y además se dice que* G_1 *es sub-isomorfo a* G_2*.*

Definición 4 (Soporte). *Siendo* $D = \{G_1, \ldots, G_{|D|}\}$ *una colección de grafos etiquetados y* G *un grafo etiquetado en* L*, el valor del* soporte *de* G *en* D *se define como el conjunto de grafos* $G_i \in D$*, tal que exista un sub-isomorfismo entre* G *y* G_i*. Este valor de soporte se obtiene mediante la ecuación (1):*

$$supp(G, D) = \frac{|\{G_i \in D\text{: } G \text{ es sub-isomorfo a } G_i\}|}{|D|}. \tag{1}$$

Definición 5 (Semejanza). *Siendo Ω el conjunto de todos los posibles grafos etiquetados en el dominio de todas las posibles etiquetas L, la semejanza entre dos grafos $G_1, G_2 \in \Omega$ se define como una función $sim : \Omega \times \Omega \to [0, 1]$. Se dice que los grafos son diferentes si $sim(G_1, G_2) = 0$, mientras mayor sea el valor de $sim(G_1, G_2)$ más semejantes son los grafos, y si $sim(G_1, G_2) = 1$ entonces existe un isomorfismo entre los grafos.*

Definición 6 (Isomorfismo aproximado y sub-isomorfismo aproximado). *Siendo G_1, G_2 y G_3 tres grafos y τ el umbral de mínima semejanza, se dice que existe un* isomorfismo aproximado *entre G_1 y G_2 si $sim(G_1, G_2) \geq \tau$. Si existe un isomorfismo aproximado entre G_1 y G_2, y $G_2 \subseteq G_3$, entonces se dice que G_1 es* sub-isomorfo aproximado *a G_3, denotado por $G_1 \subseteq_A G_3$.*

Definición 7 (Ocurrencia y conjunto de ocurrencias). *Dados los grafos $G_1 = (V_1, E_1, I_1, J_1)$, $G_2 = (V_2, E_2, I_2, J_2)$ y $T = (V_T, E_T, I_T, J_T)$, donde $T \subseteq G_2$. Se dice que T es una* ocurrencia *de G_1 en G_2 si $G_1 \subseteq_A T, |V_1| = |V_T|$ y $|E_1| = |E_T|$. El* conjunto de ocurrencias *de G_1 en G_2 se denota por $O(G_1, G_2)$.*

Definición 8 (Extensión, extensión hacia delante y hacia atrás). *Dados dos grafos $G_1 = (V_1, E_1, I_1, J_1)$ y $G_2 = (V_2, E_2, I_2, J_2)$, donde $G_1 \subseteq G_2$, se dice que $e = \{u, v\} \in E_2$ es una* extensión *de G_1 si: $V_2 = V_1 \cup \{v\}$ y $E_1 = E_2 \setminus \{e\}$, denotado por $G_2 = G_1 \diamond e$. Se dice que e es una* extensión hacia atrás *(backward extension) si $v \in V_1$, en otro caso se dice que es una* extensión hacia delante *(forward extension) si extiende el conjunto de vértices de G_1.*

Definición 9 (Conjunto de extensiones). *Siendo T una ocurrencia de G_1 en $G_2 = (V_2, E_2, I_2, J_2)$. Entonces el* conjunto de extensiones *de T se define como $ExtSet(T) = \{e | e \in E_2$, e es una extensión de $T\}$.*

Debido a que entre dos grafos etiquetados puede existir más de una correspondencia entre sus vértices y aristas, es necesario definir la semejanza máxima entre dos grafos.

Definición 10 (Semejanza máxima). *Siendo $S(G_1, G_2)$ el conjunto de todas las semejanzas entre dos grafos G_1 y G_2, la* semejanza máxima *se define como $sim_{max}(G_1, G_2) = max\{S(G_1, G_2)\}$, es decir, es la semejanza de mayor valor que se puede obtener entre las diferentes correspondencias entre G_1 y G_2.*

Utilizando las definiciones 6 y 10 se puede definir un soporte que permita utilizar correspondencia inexacta entre grafos.

Definición 11 (Soporte aproximado). *Sea $D = \{G_1, \ldots, G_{|D|}\}$ una colección de grafos etiquetados y G un grafo etiquetado, el valor del soporte aproximado (denotado por appSupp) de G en D, en términos de la semejanza, se obtiene mediante la ecuación (2):*

$$appSupp(G, D) = \frac{\sum_{\{G_i | G_i \in D, G \subseteq_A G_i\}} sim_{max}(G, G_i)}{|D|}. \tag{2}$$

Definición 12 (Subgrafo frecuente aproximado). *Un grafo etiquetado G es un subgrafo frecuente aproximado (SFA) en D si $appSupp(G, D) \geq \delta$ utilizando (2).*

El valor del umbral de soporte δ está entre $[0, 1]$ dado que la semejanza está definida entre $[0, 1]$.

Definición 13 (Minería de subgrafos frecuentes aproximados). *La minería de subgrafos frecuentes aproximados consiste en encontrar todos los SFA en una colección de grafos etiquetados D, utilizando una función de semejanza $sim_{max}(G_1, G_2)$ y un umbral de soporte δ.*

2.2. Síntesis y conclusiones

En esta sección se han definido formalmente los conceptos que son necesarios para un mejor entendimiento del resto del documento. Primero se presentaron los conceptos básicos de la teoría de grafos que caracterizan el problema de la minería de grafos. Además, se presentaron algunas definiciones importantes para la MSFA como soporte aproximado y subgrafo frecuente aproximado. Todos estos conceptos son utilizados como base para la definición de los algoritmos del estado del arte que se presentan en la sección 3, así como un lenguaje común que será utilizado en el resto de este documento.

3. Trabajos relacionados

En la literatura se han reportado varios algoritmos para la MSFA en colecciones de grafos, los cuales usan diferentes funciones de similaridad en la correspondencia entre grafos. Existen varios enfoques para la MSFA, por ejemplo:

- Algoritmos basados en distancia de edición [6,24], donde todos los posibles caminos de edición de un grafo son explorados durante el proceso de generación de los candidatos. En el algoritmo *SUBDUE* [6] se buscan sub-estructuras frecuentes en un solo grafo mediante la identificación de los caminos de menor costo explorados, mientras que el algoritmo *RNGV* [24] no busca el camino de menor costo, solamente busca uno que satisfaga la inexactitud especificada.
- Algoritmos basados en β-arista sub-isomorfismo [25,28], el cual solamente permite distorsiones entre aristas y etiquetas de aristas.
- Algoritmos basados en sub-homeomorfismo con vértices/aristas disjuntas [23,29], los cuales calculan estructuras aproximadas con topología invariante.
- Algoritmos basados en sub-isomorfismo entre grafos inciertos [30,31,32], donde el soporte esperado para cada candidato es calculado sobre una colección de subgrafos construida utilizando las probabilidades de que no ocurran en la colección original.
- Algoritmos basados en probabilidades de sustitución [15,20,21,22,27], donde no siempre una etiqueta de vértice o una etiqueta de arista puede reemplazar o ser reemplazada por otra. El algoritmo *gApprox* [22] está desarrollado para procesar un solo grafo, mientras que los algoritmos *VEAM* [15,27] y *APGM* [20,21] utilizan matrices de sustitución para realizar la MSFA en colecciones de grafos, preservando la topología de los grafos. En APGM, solamente se tratan las variaciones entre etiquetas de vértices mientras que en VEAM se permiten variaciones entre etiquetas de vértices y aristas.

Los trabajos mencionados anteriormente han sido aplicados en diferentes dominios tales como: análisis de estructuras bioquímicas [20,21,23,25,28,29,30,31], análisis de redes genéticas regulatorias [24], análisis de redes sociales y de vínculos [6], entre otros. Sin embargo, solo unos pocos trabajos han sido aplicados en tareas de clasificación de imágenes [15,16,18,19,33]. Estos últimos trabajos han reportado mejores resultados que los trabajos basados en minería exacta, destacándose que los patrones obtenidos por VEAM han mostrado ser los mejores en tareas de clasificación de imágenes.

3.1. Un método aproximado para la MSFA

El algoritmo VEAM (***V**ertex and **E**dge **A**pproximate graph **M**iner*) introduce las aproximaciones entre las etiquetas de las aristas y los vértices utilizando matrices de sustitución [15]. Como se mencionó anteriormente, dichas aproximaciones se tratan manteniendo la topología de los grafos.

Definición 14 (Matriz de sustitución y matriz de sustitución estable). *Una* matriz de sustitución $M = (m_{i,j})$ *es una matriz* $|L| \times |L|$ *indizada por el conjunto de etiqueta* L*, donde una celda* $m_{i,j}$ *en* M $(0 \leq m_{i,j} \leq 1, \sum_j m_{i,j} = 1)$ *corresponde a la probabilidad de que la etiqueta* i *sea reemplazada por la etiqueta* j*. Si* M *es diagonal dominante (i.e.* $m_{i,i} \geq m_{i,j}, \forall i \neq j$*) entonces se dice que* M *es una* matriz de sustitución estable*.*

Para calcular las aproximaciones tratadas en VEAM se utilizan dos matrices: una matriz de sustitución para las etiquetas de las aristas y otra para las etiquetas de los vértices son utilizadas, según la definición 16.

Definición 15 (Isomorfismo aproximado utilizando matrices de sustitución). *Sean* $G_1 = (V_1, E_1, I_1, J_1)$ *y* $G_2 = (V_2, E_2, I_2, J_2)$ *dos grafos etiquetados en* L*, siendo* MV *y* ME *las matrices de sustitución indizadas por* L_V *y* L_E *respectivamente, y* τ *el umbral de mínimo isomorfismo, se dice que* G_1 *es* isomorfo aproximado *a* G_2 *si:*

$$S_h(G_1, G_2) = \prod_{u \in V_1} \frac{MV_{I_1(u), I_2(h(u))}}{MV_{I_1(u), I_1(u)}} * \prod_{e=\{u,v\} \in E_1} \frac{ME_{J_1(e), J_2(\{h(u), h(v)\})}}{ME_{J_1(e), J_1(e)}} \geq \tau. \tag{3}$$

Definición 16 (Sub-isomorfismo aproximado utilizando matrices de sustitución). *Sean* $G_1 = (V_1, E_1, I_1, J_1)$*,* $G_2 = (V_2, E_2, I_2, J_2)$ *y* $G_3 = (V_3, E_3, I_3, J_3)$ *tres grafos etiquetados en* L*, donde* $G_3 \subseteq G_1$*, siendo* MV *y* ME *las matrices de sustitución indizadas por* L_V *y* L_E *respectivamente, y* τ *el umbral de mínimo isomorfismo, se dice que* G_2 *es* sub-isomorfo aproximado *a* G_1*, denotado por* $G_2 \sqsubseteq_A G_1$*, si* G_2 *es isomorfo aproximado a* G_3 *según la definición 15.*

El producto normalizado de las probabilidades de sustitución de h, denotado por $S_h(G_1, G_2)$, se conoce como el *grado de sub-isomorfismo aproximado*. En un grafo pueden encontrarse más de una ocurrencia de un subgrafo que cumpla con τ según la definición 16. Por lo que, para realizar el conteo del soporte, se utiliza la ocurrencia con mayor grado de sub-isomorfismo aproximado, denotada por $S_{max}(G_1, G_2)$, aunque se utilicen todas (las que cumplan con τ) para realizar el crecimiento del patrón en el proceso de la minería.

Una vez presentadas las definiciones anteriores, entonces se muestra el algoritmo VEAM mediante tres procedimientos principales (ver algoritmos 1, 2 y 3). Básicamente, este algoritmo explora el conjunto de grafos partiendo de los vértices frecuentes aproximados de una colección de grafos dada (ver procedimiento *Main*).

Mediante el procedimiento *Search* se realiza una búsqueda recursiva, donde se crece en profundidad un patrón dado tomando en cuenta solo el conjunto de ocurrencias (ver definición 7) y el conjunto de extensiones en los grafos de la colección (ver definición 9) de dicho patrón a crecer. Luego, se construyen los candidatos utilizando los conjuntos de posibles etiquetas, obtenidas por el procedimiento *appLSet*, con las que se obtienen grafos aproximados. De este conjunto de candidatos aproximados se almacenan y se continúan creciendo los que cumplan con el soporte aproximado dado un δ (ver definición 11) y que no hayan sido calculados en búsquedas anteriores. De esta manera se obtienen todos los SFAs de la colección de grafos dada, siendo éstos la respuesta del algoritmo VEAM para la MSFA.

Algoritmo 1: Main (D, MV, ME, δ, τ, F)

Input: D : Colección de grafos, MV : Matriz de sustitución indizada por L_V, ME : Matriz indizada por L_E, τ : Umbral de mínimo isomorfismo, δ : Umbral de frecuencia mínima.

Output: F : Conjunto de subgrafos frecuentes aproximados.

```
1 F ← C ← {Conjunto de todos los vértices etiquetados en L_V que son frecuentes aproximados en D};
2 forall T ∈ C do
3     Search(T, D, MV, ME, δ, τ, F);
```

Algoritmo 2: Search (T, D, MV, ME, δ, τ, F)

Input: $T = (V_t, E_t, I_t, J_t)$: Un subgrafo aproximado frecuente, D : Colección de grafos, MV : Matriz de sustitución indizada por L_V, ME : Matriz indizada por L_E, τ : Umbral de mínimo isomorfismo, δ : Umbral de frecuencia mínima.

Output: F : Conjunto de subgrafos frecuentes aproximados.

```
1  forall o_j ∈ O(T; G_i), donde G_i ∈ D do
2      forall e = {u, v}, e ∈ ExtSet(o_j) do
3          CL ←appLSet(T, MV, ME, G_i, o_j, e, τ);
4          forall (eLabel, vLabel) ∈ CL do
5              Se construye el candidato X utilizando la tupla (eLabel, vLabel);
6              Se calcula el código CAM de X y se almacena en codeCAM(X);
7              C ← C ∪ {(X, codeCAM(X), score)};
8  forall T_1 ∈ C do
9      if appSupp(T_1, D) ≥ δ y codeCAM(T_1) ∉ F then
10         Se inserta T_1 en F;
11         Search(T, D, MV, ME, δ, τ, F);
```

El procedimiento *appLSet* mostrado en el algoritmo 3 es el encargado de calcular la semejanza entre los candidatos y sus ocurrencias utilizando la definición 16. De esta manera se calcula el conjunto de posibles etiquetas a utilizar para la confección de los candidatos aproximados por cada grafo de la colección.

Algoritmo 3: appLSet (T, MV, ME, G, G', e, τ)

Input: T : Un grafo candidato, MV : Matriz de sustitución indizada por L_V, ME : Matriz indizada por L_E, G : Un grafo de la colección, G' : Embebido de T en G, $e = \{u, v\}$: Una extensión de G', τ : Umbral de mínimo isomorfismo.
Output: CL : Conjunto de tuplas candidatas $(eLabel, vLabel)$.

```
1 forall j ∈ L_E do
2     scoreE ← S_max(T, G') * ME_{j,J(e)} / ME_{j,j};
3     if e es una extensión hacia delante de G' then
4         forall i ∈ L_V do
5             score ← scoreE * MV_{i,I(v)} / MV_{i,i};
6             if score ≥ τ then CL ← CL ∪ {(j, i)};
7     else if scoreE ≥ τ then CL ← CL ∪ {(j, ∅)}
```

Por otro lado, en el algoritmo *APGM* [21] solamente se utiliza el producto normalizado de las probabilidades de sustitución entre las etiquetas de los vértices. Por lo que el sub-isomorfismo aproximado utilizando matrices de sustitución para APGM se calcula mesiante la ecuacuación (4).

$$S'_h(G_1, G_2) = \prod_{u \in V_1} \frac{MV_{I_1(u), I_2(h(u))}}{MV_{I_1(u), I_1(u)}} \geq \tau. \tag{4}$$

Algoritmo 4: appLSet-APGM (T, MV, G, G', e, τ)

Input: T : Un grafo candidato, MV : Matriz de sustitución indizada por L_V, G : Un grafo de la colección, G' : Embebido de T en G, $e = \{u, v\}$: Una extensión de G', τ : Umbral de mínimo isomorfismo.
Output: CL : Conjunto de tuplas candidatas $(eLabel, vLabel)$.

```
1 if e es una extensión hacia delante de G' then
2     forall i ∈ L_V do
3         score ← S_max(T, G') * MV_{i,I(v)} / MV_{i,i};
4         if score ≥ τ then CL ← CL ∪ {(J(e), i)};
5 else CL ← CL ∪ {(J(e), ∅)}
```

Finalmente, el algoritmo APGM se puede conformar sustituyendo el procedimiento *appLSet* por el procedimiento *appLSet-APGM* en la línea 3 del procedimiento mostrado mediante el algoritmo 2, manteniendo el resto de los procedimientos presentados para VEAM. El procedimiento appLSet-APGM mostrado en el algoritmo 4 es el encargado de calcular la semejanza entre los candidatos y sus ocurrencias utilizando la ecuación (4). De esta manera se calcula el conjunto de posibles etiquetas de los vértices a utilizar para la confección de los candidatos aproximados por cada grafo de la colección.

3.2. Mejoras propuestas para la MSFA

En el proceso de MSFA se realiza un elevado número de pruebas de sub-isomorfismo y de formas canónicas comparado con el proceso de minería de los algoritmos exactos. En este tipo de minería no se pueden hacer cortes de aristas en los grafos luego de ser procesados de forma directa ya que pueden ser ocurrencias aproximadas de otros candidatos. Estos cortes en los algoritmos exactos evitan procesamientos duplicados en los candidatos. Es por este motivo que Acosta-Mendoza *et al.* [27] presentan una mejora para los algoritmos de MSFA mediante dos podas enfocadas a la reducción del espacio de búsqueda de las etiquetas de las aristas y los vértices. Estas podas permiten evitar un gran número de pruebas de sub-isomorfismo y de formas canónicas en posibles candidatos que nunca serían seleccionados por el proceso de la minería al no tener posibilidades de cumplir con el umbral de mínimo isomorfismo τ.

Definición 17 (Conjunto de etiquetas útiles). *Sean $l_v \in L_V$, $l_e \in L_E$ una etiqueta de vértice y una etiqueta de arista, respectivamente, se dice que los conjuntos de etiquetas, para l_v y l_e, son útiles si son obtenidas mediante las funciones $U_V^\tau : L_V \to P_{L_V}$ y $U_E^\tau : L_E \to P_{L_E}$, respectivamente, tal que:*

- $U_V^\tau(l_v) = \{l | l \in L_V, \frac{MV_{l,l_v}}{MV_{l,l}} \geq \tau\}$,
- $U_E^\tau(l_e) = \{l | l \in L_E, \frac{ME_{l,l_e}}{ME_{l,l}} \geq \tau\}$;

donde, P_{L_V} y P_{L_E} son los conjuntos conocidos como power sets [1] *de L_V y L_E, respectivamente.*

Teniendo en cuenta la definición 17, Acosta-Mendoza *et al.* [27] presentaron algunos teoremas basados en esta definición que justifican teóricamente las podas propuestas. Finalmente, los algoritmos VEAM y APGM haciendo uso de las podas propuestas quedaron como se muestra mediante los algoritmos 5-8.

En el procedimiento *Improved-Main*, mostrado mediante el algoritmo 5, se realiza la primera reducción del espacio de búsqueda de la colección. En las líneas 2 y 5 se eliminan los vértices y aristas de la colección que no serán utilizados como ocurrencias por ningún subgrafo frecuente en el resto del proceso de crecimiento de patrones.

Algoritmo 5: Improved-Main (D, MV, ME, δ, τ, F)

Input: D : Colección de grafos, MV : Matriz de sustitución indizada por L_V, ME : Matriz indizada por L_E, τ : Umbral de mínimo isomorfismo, δ : Umbral de frecuencia mínima.
Output: F : Conjunto de subgrafos frecuentes aproximados.

1 $F \leftarrow$ {Conjunto de todos los vértices etiquetados en L_V que son frecuentes aproximados en D};
2 Se eliminan de D los vértices con la etiqueta l_v tal que $L_U^V \cap U_V^\tau(l_v) = \emptyset$;
3 $C \leftarrow$ {Conjunto de todas las aristas etiquetadas en L_E que son frecuentes aproximadas en D};
4 $F \leftarrow F \cup C$;
5 Se eliminan de D las aristas con la etiqueta l_e tal que $L_U^E \cap U_E^\tau(l_e) = \emptyset$;
6 **forall** $T \in C$ **do**
7 `Improved-Search`(T, D, MV, ME, δ, τ, F);

El procedimiento *Search* es igual al mostrado en la sección 3.1, mediante el cual se realiza una búsqueda recursiva, donde se crece en profundidad un patrón dado tomando en cuenta solo el

[1] Para un conjunto X, el power set de X es $P_X = \{Y | Y \subseteq X\}$

conjunto de ocurrencias (ver definición 7) y el conjunto de extensiones en los grafos de la colección (ver definición 9) de dicho patrón a crecer. Luego, se construyen los candidatos utilizando los conjuntos de posibles etiquetas, obtenidas por el procedimiento *appLSet*, con las que se obtienen grafos aproximados. De este conjunto de candidatos aproximados se almacenan y se continúan creciendo los que cumplan con el soporte aproximado dado un δ (ver definición 11) y que no hayan sido calculados en búsquedas anteriores. De esta manera se obtienen todos los SFAs de la colección de grafos dada, siendo éstos la respuesta del algoritmo VEAM para la MSFA.

Algoritmo 6: Search (T, D, MV, ME, δ, τ, F)

Input: $T = (V_t, E_t, I_t, J_t)$: Un subgrafo aproximado frecuente, D : Colección de grafos, MV : Matriz de sustitución indizada por L_V, ME : Matriz indizada por L_E, τ : Umbral de mínimo isomorfismo, δ : Umbral de frecuencia mínima.

Output: F : Conjunto de subgrafos frecuentes aproximados.

```
1  forall o_j ∈ O(T; G_i), donde G_i ∈ D do
2      forall e ∈ ExtSet(o_j) do
3          CL ← Improved-appLSet(T, MV, ME, G_i, o_j, e, τ);
4          forall (eLabel, vLabel) ∈ CL do
5              Se construye el candidato X utilizando la tupla (eLabel, vLabel);
6              Se calcula el código CAM de X y se almacena en codeCAM(X);
7              C ← C ∪ {(X, codeCAM(X), score)};
8  forall T_1 ∈ C do
9      if appSupp(T_1, D) ≥ δ y codeCAM(T_1) ∉ F then
10         Se inserta T_1 en F;
11         Improved-Search(T, D, MV, ME, δ, τ, F);
```

Algoritmo 7: Improved-appLSet (T, MV, ME, G, G', e, τ)

Input: T : Un grafo candidato, MV : Matriz de sustitución indizada por L_V, ME : Matriz indizada por L_E, G : Un grafo de la colección, G' : Embebido de T en G, $e = \{u, v\}$: Una extensión de G', τ : Umbral de mínimo isomorfismo.

Output: CL : Conjunto de tuplas candidatas $(eLabel, vLabel)$.

```
1  forall j ∈ U_E^τ(J(e)) do
2      scoreE ← S_max(T, G') * ME_{j,J(e)} / ME_{j,j};
3      if e es una extensión hacia delante de G' then
4          forall i ∈ U_V^τ(I(v)) do
5              if i es menor o igual que la mayor etiqueta de vértices de T then
6                  score ← scoreE * MV_{i,I(v)} / MV_{i,i};
7                  if score ≥ τ then CL ← CL ∪ {(j, i)};
8      else if scoreE ≥ τ then CL ← CL ∪ {(j, ∅)}
```

Los procedimientos *Improved-appLSet* e *Improved-appLSet-APGM* calculan la semejanza entre los candidatos y sus ocurrencias utilizando la definición 16 y la ecuación (4), respectivamente. De esta manera se calcula el conjunto de posibles etiquetas a utilizar para la confección de los candidatos aproximados por cada grafo de la colección. Este conjunto de posibles etiquetas son calculadas sobre el conjunto de etiquetas útiles, según la definición 17, en lugar de todas las etiquetas para los vértices y aristas. Esto permite disminuir el espacio de búsqueda de las eti-

quetas. El procedimiento Improved-appLSet le pertenece al algoritmo VEAM y el procedimiento Improved-appLSet-APGM de pertenece al algoritmo APGM.

Al sustituir la llamada de la línea 3 del procedimiento *Improved-Search* por una llamada al procedimiento *Improved-appLSet-APGM* se logra la confección del algoritmo APGM.

Algoritmo 8: Improved-appLSet-APGM (T, MV, G, G', e, τ)

Input: T : Un grafo candidato, MV : Matriz de sustitución indizada por L_V, G : Un grafo de la colección, G' : Embebido de T en G, $e = \{u, v\}$: Una extensión de G', τ : Umbral de mínimo isomorfismo.
Output: CL : Conjunto de tuplas candidatas $(eLabel, vLabel)$.

```
1 if e es una extensión hacia delante de G' then
2     forall i ∈ U_V^τ(I(v)) do
3         score ← S_max(T, G') * MV_{i,I(v)} / MV_{i,i};
4         if score ≥ τ then CL ← CL ∪ {(J(e), i)};
5 else CL ← CL ∪ {(J(e), ∅)}
```

Por otro lado, a pesar de que los SFAs calculados por VEAM han mostrado ser útiles para la clasificación de imágenes, se detectó que dicho algoritmo calcula un gran número de patrones en el proceso de la minería. Muchos de estos patrones calculados no aportan información útil para la clasificación, puesto que no se verifica que son representativos para alguna clase en específico y por lo tanto no son de utilidad como atributos para la clasificación. Además, el número de patrones crece a medida que disminuye el umbral de soporte y/o el umbral de mínimo isomorfismo, lo cual afecta negativamente en el rendimiento de los clasificadores. Por ello, se ha propuesto la integración de un módulo de selección de atributos en el esquema de clasificación donde se utilizan los SFAs calculados por VEAM como atributos para la clasificación [19]. Dicha selección no solo permitió una reducción de la dimensionalidad del conjunto de atributos a utilizar en la clasificación sino que se lograron mejoras en los resultados de la clasificación.

3.3. Síntesis y conclusiones

En esta sección se han descrito los trabajos relacionados con esta investigación. Se ha hecho énfasis en los algoritmos para la MSFA, específicamente en el algoritmo VEAM. Finalmente, se comentaron algunas mejoras reportadas para el proceso de la MSFA, las cuales permitieron mejorar tanto la eficiencia de la MSFA como la eficacia en el uso de los SFA calculados.

4. Resultados alcanzados utilizando la MSFA

Como se mencionó anteriormente, la MSFA se ha utilizado en tareas de clasificación de imágenes donde se han obtenido buenos resultados. En esta sección se muestran dichos resultados sobre varias colecciones de imágenes sintéticas y reales representadas en forma de grafos. Además, se mencionan varias podas para la MSFA con el fin de acelerar el proceso de minería, así como un módulo de selección de atributos que se integró en el esquema de clasificación de imágenes, utilizando los subgrafos frecuentes aproximados (SFAs) calculados por un algoritmo para la MSFA, con el objetivo de disminuir la dimensionalidad del conjunto de atributos utilizados en la clasificación.

4.1. Colecciones de grafos utilizadas

Varias colecciones de imágenes han sido utilizadas para probar el enfoque de los algoritmos para la MSFA, en específico el enfoque de VEAM [15]:

- COIL-100 [34], donde se tienen imágenes de objetos reales tomados desde diferentes puntos de vistas.
- GREC [35], donde las imágenes representan símbolos de los planos arquitectónicos o electrónicos.
- Imágenes sintéticas obtenidas mediante el Generador aleatorio de imágenes de Coenen[2], donde las imágenes representan dos tipos de vistas (marítimas y terrestres).

En el caso de la colección COIL se utilizan 25 objetos aleatorios de 100 que posee la colección. Además, la colección de imágenes sintéticas (*CoenenDB*) está compuesta por 2000 imágenes. Cada colección está dividida aleatoriamente en dos sub-colecciones:

- COIL: se divide en 198 (11 %) imágenes para el entrenamiento y 1602 para prueba.
- GREC: se divide en 572 (52 %) imágenes para el entrenamiento y 528 para prueba.
- CoenenDB: se divide en 1200 (60 %) imágenes para el entrenamiento y 800 para prueba.

Las características específicas de cada colección se muestran en la tabla 1.

Tabla 1. Colecciones de imágenes utilizadas.

Colección	**COIL**	**GREC**	**CoenenDB**
Cantidad grafos	1800	1100	2000
Cantidad etiquetas de vértices	152	4	18
Cantidad etiquetas de aristas	27	24	24
Tamaño promedio de los grafos	135	11	49
Cantidad clases	25	22	2

Las imágenes de las colecciones mencionadas se representan en forma de grafos no dirigidos y etiquetados utilizando diferentes algoritmos de representación como:

- En COIL se representaron haciendo uso de las pirámides irregulares de grafos de cada imagen que proveen una jerarquía de las particiones a diferentes niveles de resolución [36,37] y seleccionando el grafo de mejor calidad utilizando una la medida propuesta por Morales-González y García-Reyes [38].
- Coenen images se representaron utilizando la información de las hojas del árbol generado mediante la técnica de quad-tree [39].
- GREC se representa en forma de grafos utilizando los puntos críticos en las imágenes seleccionados de forma semiautomática presentado por Riesen y Bunke [35].

4.2. Clasificación de imágenes

La MSFA se ha podido aplicar en tareas de clasificación de imágenes gracias a los esfuerzos de algunos investigadores que han desarrollado técnicas para la representación de estas imágenes en

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

forma de grafos [35,39,38]. Mediante este tipo de representación se puede describir la información estructural y topológica de las imágenes permitiendo convertir la tarea de clasificación de imágenes en una de clasificación de grafos.

El problema de clasificación de imágenes que se trata en este trabajo se inicia con la representación en forma de grafos de un conjunto de imágenes pre-etiquetadas, seguido de la confección de las matrices de sustitución usadas en el modelo aproximado. De esta manera se pueden calcular los SFAs mediante un algoritmo para la MSFA (VEAM) utilizando un umbral de soporte mínimo ($0 < \delta \leq 1$) y un umbral de mínimo isomorfismo ($0 < \tau \leq 1$). Estos dos argumentos son utilizados para reducir el espacio de búsqueda en el proceso de minería. Luego, partiendo de los patrones (SFAs) calculados se construyen los vectores de características que representarán a las imágenes y son necesarios para los algoritmos de clasificación. De esta manera se construye una matriz donde el número de filas ($1 \leq i \leq |D|$) es el número de imágenes de la colección y el número de columnas ($1 \leq j \leq x$) es el número de patrones calculados por VEAM. Finalmente estos vectores de atributos son utilizados por el clasificador para lograr la clasificación de las imágenes.

Las imágenes de la colección de prueba son representadas en forma de grafos de la misma manera que el conjunto de entrenamiento. El conjunto de grafos de prueba se representa mediante vectores de atributos haciendo uso de los patrones calculados en la fase de entrenamiento y teniendo en cuenta la existencia aproximada de cada uno de estos patrones en los grafos. Finalmente, se realiza la clasificación del conjunto de imágenes de prueba utilizando el modelo del entrenamiento y los vectores que representan cada imagen de dicho conjunto. De esta manera queda descrito el proceso general de clasificación de imágenes mostrado en la figura 1.

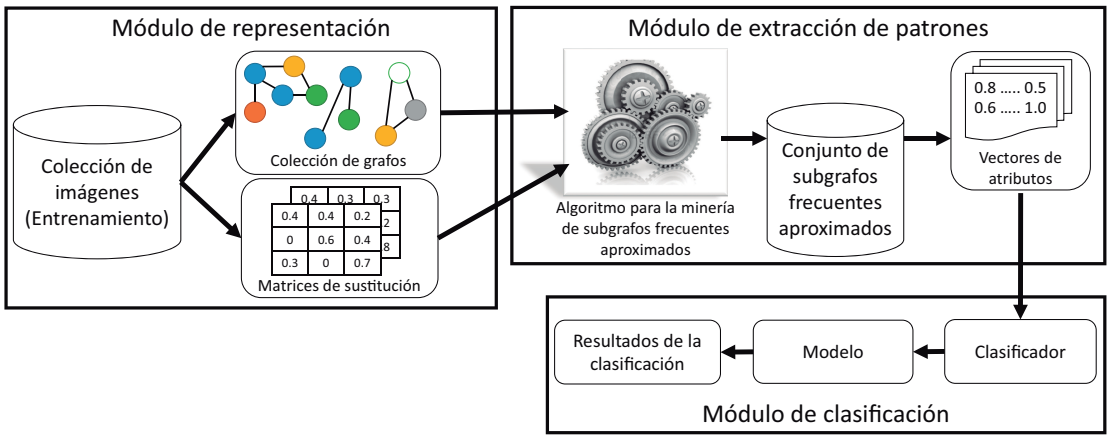

Fig. 1. Esquema de clasificación de imágenes basadas en grafos.

El esquema de la figura 1 representa la base de todo método de clasificación de imágenes que utilice MSFA. En este esquema pueden variar la representación de las imágenes, la confección de las matrices de sustitución, los algoritmos para la minería y los clasificadores a utilizar; sin embargo, la esencia de este esquema seguirá siendo la ilustrada en la figura 1. Por lo que este esquema fue utilizado como base en diferentes trabajos que realizan tareas de clasificación de imágenes [15,16,17,18], donde en algunos se construye las matrices de sustitución de forma automática mediante técnicas de agrupamiento sobre las regiones de las imágenes con características similares [16,18]. En los trabajos mencionados se han alcanzado buenos resultados utilizando, principalmente, el clasificador SVM (de sus siglas en inglés, Support Vector Machine) sobre varias colecciones de imágenes.

Los primeros avances y motivación de la aplicación de la MSFA en tareas de clasificación de imágenes fueron logrados sobre colecciones sintéticas de grafos obtenidas mediante el Generador de imágenes aleatorias de Coenen (ver resultados reportados por Acosta-Mendoza *et al.* [15]). En este trabajo se utiliza una colección diferente con mayor número de imágenes obtenida mediante el mismo generador. En este caso, también se puede observar que los patrones calculados por VEAM son de utilidad para la clasificación. Esto se debe a que tener en cuenta las distorsiones en las aristas es importante en la clasificación de imágenes ya que los objetos tienden a tener variaciones en las imágenes de una misma clase. Además, se muestran los resultados obtenidos sobre varias colecciones de imágenes reales como GREC, donde el algoritmo APGM calcula los mismos patrones que los algoritmos exactos ya que no existen semejanzas semánticas en los vértices de dicha colección. Además, se muestran los resultados sobre la colección COIL-100. Estas colecciones han sido las aplicaciones más recientes de la MSFA y tienen un nivel mayor de complejidad respecto a las demás colecciones descritas anteriormente.

A continuación en las tablas 2 y 3 se muestran los resultados de la clasificación obtenidos en las colecciones descritas en la sección 4.1 utilizando el esquema de clasificación antes presentado en la figura 1. En estas tablas se realiza una comparación entre los algoritmos exactos y los algoritmos para la MSFA (APGM y VEAM) con el objetivo de mostrar la utilidad de los patrones calculados por la MSFA, específicamente los calculados por VEAM.

La primera columna de las tablas 2 y 3 indican la colección usada y la segunda columna especifica el umbral de soporte mínimo. Luego, las nueve columnas siguientes se agrupan en tres tríos, donde cada trío muestra los resultados de la clasificación obtenidos por el clasificador indicado en la parte superior de dichas tablas.

Como se puede observar las tablas 2 y 3, los algoritmos para la MSFA obtienen mejores resultados que los métodos exactos en la mayoría de los casos. Por otro lado, en el caso de los métodos aproximados se puede observar que en general los SFA calculados por VEAM obtiene mejores resultados que los SFA calculados por APGM. Esto último muestra que el tratamiento de las distorsiones en las aristas provee información relevante y adicional para la clasificación. Esto permite que los SFA calculados por VEAM obtengan mejores resultados en la mayoría de los casos. Finalmente, estos resultados muestran que los patrones calculados por VEAM son más viables para estas tareas de clasificación que los calculados por APGM o los algoritmos exactos.

Adicionalmente, en la tabla 4 se presentan comparaciones estadísticas entre los clasificadores en las diferentes colecciones utilizando los algoritmos para la minería. Para esta comparación se utiliza una prueba de significancia estadística reportada por García y Herrera [40] conocida como *Bergmann* [41]. Para las pruebas de significancia realizadas en este trabajo se utilizaron 0.5 y 0.10 como valores de α.

Como se puede observar en la tabla 4 existe significancia estadística entre los resultados de clasificación que usan patrones calculados por algoritmos exactos y los resultados de clasificación que usan patrones calculados por algoritmos aproximados, identificándose estos últimos como mejor opción para calcular patrones que caractericen a las colecciones de imágenes utilizadas en los experimentos. Además, de forma general, los patrones calculados por VEAM presentan un mejor comportamiento en la clasificación que los patrones calculados por APGM. Por otro lado, es posible realizar una comparación del método que utiliza la MSFA con métodos que utilizan otras técnicas de clasificación que no son basadas en la minería. En COIL, con el método propuesto por Morales-González y García-Reyes [38], que no está basado en MSFA, se obtiene 91.60 % de accuracy mientras que con los métodos basados en la MSFA se obtiene 91.32 % de accuracy y 92.19 % de F-measure.

Tabla 2. Resultados de la clasificación (accuracy) utilizando varios clasificadores sobre diferentes colecciones de grafos.

		J48graft			Decision Table			Regression		
Colección	δ	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**
	20 %	95.38 %	96.00 %	**97.25** %	89.63 %	91.13 %	**94.38** %	95.63 %	96.13 %	**96.25** %
	25 %	95.50 %	**96.88** %	96.75 %	**92.50** %	92.25 %	80.13 %	95.25 %	**96.38** %	**96.38** %
	30 %	95.38 %	**96.50** %	**96.50** %	92.38 %	93.25 %	**95.25** %	95.75 %	**96.75** %	96.50 %
CoenenDB	35 %	95.00 %	95.38 %	**96.88** %	92.63 %	92.63 %	**94.38** %	94.13 %	94.13 %	**96.88** %
	40 %	79.75 %	79.88 %	**93.50** %	78.63 %	78.63 %	**93.00** %	77.13 %	78.75 %	**93.38** %
	45 %	78.88 %	78.88 %	**84.50** %	78.63 %	78.63 %	**82.75** %	77.13 %	77.75 %	**83.88** %
	50 %	78.88 %	78.88 %	**83.50** %	78.63 %	78.63 %	**82.75** %	77.13 %	77.75 %	**82.75** %
	55 %	78.88 %	78.88 %	**80.00** %	**78.63** %	**78.63** %	77.75 %	77.13 %	**77.75** %	**77.75** %
	60 %	77.75 %	**78.00** %	77.75 %	77.63 %	77.63 %	**77.88** %	77.13 %	77.13 %	77.13 %
	2 %	**64.02** %		45.45 %	**52.46** %		33.90 %	68.18 %		**73.48** %
	3 %	68.75 %		**82.20** %	52.27 %		**65.72** %	69.89 %		**83.14** %
	4 %	64.39 %		**81.63** %	52.08 %		**68.37** %	68.94 %		**82.95** %
GREC	5 %	64.58 %		**79.36** %	52.27 %		**68.37** %	68.75 %		**81.63** %
	6 %	64.58 %		**79.73** %	52.27 %		**64.96** %	68.75 %		**79.73** %
	7 %	60.98 %		**76.33** %	47.35 %		**64.02** %	67.42 %		**79.36** %
	8 %	57.01 %		**78.60** %	41.67 %		**64.58** %	64.39 %		**77.08** %
	9 %	54.36 %		**78.98** %	41.67 %		**62.31** %	63.83 %		**78.79** %
	10 %	51.89 %		**77.46** %	41.67 %		**62.31** %	63.07 %		**81.44** %
	30 %	–	75.78 %	**79.96** %	–	51.19 %	**52.06** %	–	**74.53** %	74.34 %
	40 %	–	**76.72** %	71.22 %	–	**53.37** %	53.25 %	–	75.84 %	**76.15** %
COIL	50 %	–	73.28 %	73.28 %	–	54.49 %	54.49 %	–	72.97 %	72.97 %
	60 %	–	73.22 %	73.22 %	–	52.50 %	52.50 %	–	69.29 %	69.29 %
	70 %	–	58.36 %	58.36 %	–	51.56 %	51.56 %	–	67.92 %	67.92 %
	80 %	–	63.42 %	63.42 %	–	57.99 %	57.99 %	–	70.60 %	70.60 %

		AdaBoost			BayesNet			SVM		
Colección	δ	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**
	20 %	90.63 %	91.63 %	**94.00** %	**92.63** %	88.75 %	90.38 %	94.50 %	**96.38** %	95.38 %
	25 %	90.63 %	91.63 %	**92.25** %	**91.13** %	89.13 %	90.38 %	94.50 %	**95.63** %	94.38 %
	30 %	91.50 %	**92.38** %	91.75 %	91.00 %	90.50 %	**91.13** %	94.88 %	**95.50** %	95.13 %
CoenenDB	35 %	92.50 %	**92.63** %	91.00 %	91.50 %	**92.25** %	90.88 %	93.38 %	94.00 %	**95.50** %
	40 %	76.50 %	76.50 %	**91.38** %	77.00 %	77.00 %	**91.25** %	77.88 %	77.75 %	**93.13** %
	45 %	76.50 %	76.50 %	**82.00** %	77.00 %	77.00 %	**80.75** %	77.38 %	77.38 %	**83.13** %
	50 %	76.50 %	76.50 %	**81.75** %	**77.00** %	**77.00** %	76.13 %	77.38 %	77.38 %	**81.75** %
	55 %	76.50 %	76.50 %	76.50 %	**77.00** %	**77.00** %	75.38 %	77.38 %	77.38 %	**77.88** %
	60 %	76.50 %	76.50 %	76.50 %	76.88 %	76.88 %	76.88 %	77.38 %	77.38 %	**77.75** %
	2 %	–			76.33 %		**85.61** %	83.71 %		**94.13** %
	3 %	–			75.76 %		**87.88** %	80.30 %		**94.51** %
	4 %	–			75.57 %		**88.83** %	82.01 %		**94.13** %
GREC	5 %	–			73.67 %		**86.93** %	80.49 %		**92.99** %
	6 %	–			71.02 %		**85.61** %	79.55 %		**91.86** %
	7 %	–			65.34 %		**84.09** %	77.08 %		**90.15** %
	8 %	–			63.26 %		**83.90** %	74.81 %		**89.20** %
	9 %	–			58.52 %		**83.52** %	73.86 %		**88.64** %
	10 %	–			55.87 %		**82.95** %	71.59 %		**88.83** %
	30 %	–			–	**91.32** %	90.51 %	–	**91.14** %	90.20 %
	40 %	–			–	**88.45** %	87.33 %	–	82.33 %	**83.96** %
COIL	50 %	–			–	81.90 %	81.90 %	–	77.22 %	77.22 %
	60 %	–			–	82.08 %	82.08 %	–	66.98 %	66.98 %
	70 %	–			–	75.47 %	75.47 %	–	65.54 %	65.54 %
	80 %	–			–	72.91 %	72.91 %	–	62.61 %	62.61 %

Tabla 3. Resultados de la clasificación (F-measure) utilizando varios clasificadores sobre diferentes colecciones de grafos (imágenes).

		J48graft			Decision Table			Regression		
Colección	δ	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**
CoenenDB	20 %	95.43 %	95.98 %	**97.23** %	89.87 %	91.56 %	**94.49** %	95.52 %	96.08 %	**96.21** %
	25 %	95.53 %	**96.86** %	96.73 %	**92.57** %	92.25 %	82.51 %	95.17 %	**96.37** %	96.33 %
	30 %	95.30 %	96.45 %	**96.46** %	92.46 %	93.51 %	**95.33** %	95.67 %	**96.75** %	96.50 %
	35 %	95.06 %	95.36 %	**96.87** %	92.54 %	92.54 %	**94.45** %	94.07 %	93.97 %	**96.87** %
	40 %	75.89 %	76.01 %	**93.53** %	74.52 %	74.52 %	**93.03** %	70.34 %	73.19 %	**93.37** %
	45 %	74.74 %	74.74 %	**83.47** %	74.52 %	74.52 %	**81.04** %	70.34 %	72.45 %	**85.59** %
	50 %	74.74 %	74.74 %	**81.56** %	74.52 %	74.52 %	**81.04** %	70.34 %	72.45 %	**81.04** %
	55 %	74.74 %	74.74 %	**76.19** %	**74.52** %	**74.52** %	72.27 %	70.34 %	72.45 %	**72.53** %
	60 %	72.45 %	72.45 %	72.45 %	72.33 %	72.33 %	**72.47** %	70.34 %	70.34 %	70.34 %
GREC	2 %	43.64 %		–	**17.89** %		11.76 %	41.67 %		**79.17** %
	3 %	53.57 %		**86.96** %	20.00 %		**28.13** %	41.67 %		**78.43** %
	4 %	51.85 %		**78.43** %	19.47 %		**34.29** %	41.67 %		**76.00** %
	5 %	52.00 %		**78.43** %	19.23 %		**36.00** %	40.00 %		**76.60** %
	6 %	40.00 %		**73.47** %	17.09 %		**33.96** %	44.90 %		**72.34** %
	7 %	51.61 %		**80.00** %	20.62 %		**29.75** %	41.67 %		**71.11** %
	8 %	31.37 %		**70.83** %	16.51 %		**26.02** %	40.00 %		**76.60** %
	9 %	37.74 %		**70.83** %	16.51 %		**32.20** %	42.55 %		**75.47** %
	10 %	40.00 %		**69.77** %	16.51 %		**32.20** %	42.55 %		**71.70** %
COIL	30 %	–	81.08 %	**91.18** %	–	**54.60** %	48.80 %	–	65.09 %	**67.06** %
	40 %	–	**87.32** %	75.18 %	–	55.00 %	55.00 %	–	64.41 %	**67.86** %
	50 %	–	72.00 %	72.00 %	–	57.50 %	57.50 %	–	79.75 %	79.75 %
	60 %	–	79.25 %	79.25 %	–	56.90 %	56.90 %	–	80.25 %	80.25 %
	70 %	–	28.07 %	28.07 %	–	51.70 %	51.70 %	–	75.82 %	75.82 %
	80 %	–	28.07 %	28.07 %	–	57.10 %	57.10 %	–	81.69 %	81.69 %

		AdaBoost			BayesNet			SVM		
Colección	δ	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**	**Exactos**	**APGM**	**VEAM**
CoenenDB	20 %	90.66 %	91.84 %	**93.89** %	**92.52** %	88.61 %	90.29 %	94.44 %	**96.34** %	95.39 %
	25 %	90.66 %	**91.84** %	**91.84** %	**91.02** %	89.11 %	90.34 %	94.47 %	**95.52** %	94.35 %
	30 %	91.08 %	92.11 %	**92.36** %	90.89 %	90.55 %	**91.14** %	94.76 %	**95.44** %	95.06 %
	35 %	91.29 %	**91.39** %	91.37 %	91.63 %	**92.17** %	90.84 %	93.21 %	93.81 %	**95.37** %
	40 %	80.97 %	80.97 %	**91.19** %	72.37 %	72.37 %	**91.07** %	72.56 %	72.19 %	**93.12** %
	45 %	80.97 %	80.97 %	**83.64** %	72.37 %	72.37 %	**82.66** %	72.45 %	72.45 %	**84.46** %
	50 %	**80.97** %	**80.97** %	79.72 %	72.37 %	72.37 %	**79.75** %	72.45 %	72.45 %	**79.72** %
	55 %	80.97 %	80.97 %	80.97 %	**72.37** %	**72.37** %	70.11 %	72.45 %	72.45 %	**72.64** %
	60 %	80.97 %	80.97 %	80.97 %	72.37 %	**80.13** %	79.92 %	72.45 %	72.45 %	72.45 %
GREC	2 %	–			46.15 %		**86.36** %	68.09 %		**93.33** %
	3 %	–			46.15 %		**86.96** %	58.82 %		**89.36** %
	4 %	–			47.06 %		**86.96** %	66.67 %		**86.96** %
	5 %	–			44.90 %		**82.61** %	50.00 %		**86.96** %
	6 %	–			43.14 %		**78.05** %	57.69 %		**86.36** %
	7 %	–			37.74 %		**83.72** %	58.82 %		**77.27** %
	8 %	–			37.74 %		**81.82** %	56.00 %		**80.00** %
	9 %	–			33.90 %		**80.95** %	53.85 %		**76.60** %
	10 %	–			33.90 %		**79.07** %	56.00 %		**78.26** %
COIL	30 %	–	15.54 %	15.54 %	–	83.78 %	**87.32** %	–	85.94 %	**92.19** %
	40 %	–	14.84 %	**15.54** %	–	82.89 %	**85.71** %	–	72.00 %	**84.21** %
	50 %	–	17.05 %	17.05 %	–	81.82 %	81.82 %	–	79.28 %	79.28 %
	60 %	–	11.55 %	11.55 %	–	83.44 %	83.44 %	–	38.71 %	38.71 %
	70 %	–	12.79 %	12.79 %	–	76.06 %	76.06 %	–	33.03 %	33.03 %
	80 %	–	12.79 %	12.79 %	–	80.00 %	80.00 %	–	31.03 %	31.03 %

4.3. Mejorando en eficacia

En el esquema de clasificación (ver figura 1) de la sección 4.2 lo primero que se realiza es la construcción de la colección de los grafos que representan dichas imágenes. Luego, se crean los

Tabla 4. Pruebas de significancia estadística para diferentes clasificadores en varias colecciones de grafos (imágenes) usando diferentes algoritmos para la MSFA.

(a) α =0.05

Coenen-DB						
Classifier	**J48graft**	**Decision Table**	**Regression**	**AdaBoost**	**BayesNet**	**SVM**
Exactos vs. APGM	APGM	–	APGM	–	–	–
Exactos vs. VEAM	VEAM	VEAM	VEAM	VEAM	–	VEAM
APGM vs. VEAM	VEAM	VEAM	–	–	–	–
GREC						
Classifier	**J48graft**	**Decision Table**	**Regression**	**AdaBoost**	**BayesNet**	**SVM**
Exactos vs. VEAM	VEAM	VEAM	VEAM	–	VEAM	VEAM
COIL						
Classifier	**J48graft**	**Decision Table**	**Regression**	**AdaBoost**	**BayesNet**	**SVM**
Exactos vs. APGM	APGM	APGM	APGM	–	APGM	APGM
Exactos vs. VEAM	VEAM	VEAM	VEAM	–	VEAM	VEAM
APGM vs. VEAM	–	–	–	–	–	–

(b) α =0.10

Coenen-DB						
Classifier	**J48graft**	**Decision Table**	**Regression**	**AdaBoost**	**BayesNet**	**SVM**
Exactos vs. APGM	APGM	–	APGM	APGM	–	APGM
Exactos vs. VEAM	VEAM	VEAM	VEAM	VEAM	–	VEAM
APGM vs. VEAM	VEAM	VEAM	–	–	–	–
GREC						
Classifier	**J48graft**	**Decision Table**	**Regression**	**AdaBoost**	**BayesNet**	**SVM**
Exactos vs. VEAM	VEAM	VEAM	VEAM	–	VEAM	VEAM
COIL						
Classifier	**J48graft**	**Decision Table**	**Regression**	**AdaBoost**	**BayesNet**	**SVM**
Exactos vs. APGM	APGM	APGM	APGM	–	APGM	APGM
Exactos vs. VEAM	VEAM	VEAM	VEAM	–	VEAM	VEAM
APGM vs. VEAM	–	–	–	–	–	–

vectores de atributos a partir de los patrones calculados mediante un algoritmo para la MSFA. Con dichos vectores, mediante un algoritmo de clasificación, se entrena un modelo el cual es utilizado para clasificar los vectores de atributos que representan el conjunto de imágenes de entrenamiento y de esta forma etiquetar dichas imágenes.

Mediante ese esquema de clasificación se obtuvieron buenos resultados de clasificación destacándose los resultados obtenidos por los patrones calculados por el algoritmo VEAM en el módulo de extracción de patrones (ver tablas 2 y 3). Sin embargo, el número de patrones calculados por VEAM se hace muy grande bajo algunas condiciones como: decremento de los umbrales δ y τ. En la mayoría de estos casos, algunos de los patrones calculados no aportan información relevante para los algoritmos de clasificación. Por este motivo, Acosta-Mendoza *et al.* [19] se propone un módulo de selección con el objetivo de reducir la dimensionalidad de los vectores de atributos a utilizar en la clasificación. En ese trabajo se reportan buenos resultados haciendo uso de los algoritmos de selección: ganancia de información (***I**nformation **G**ain IG*), chi-cuadrado (*CHI-Q*) y el cociente de evaluación de la ganancia de información (***G**ain **R**atio **A**ttribute **E**valuation GRAE*).

En la figura 2 se muestra el esquema de clasificación donde se incluye el módulo de selección el cual fue la contribución principal del trabajo [19].

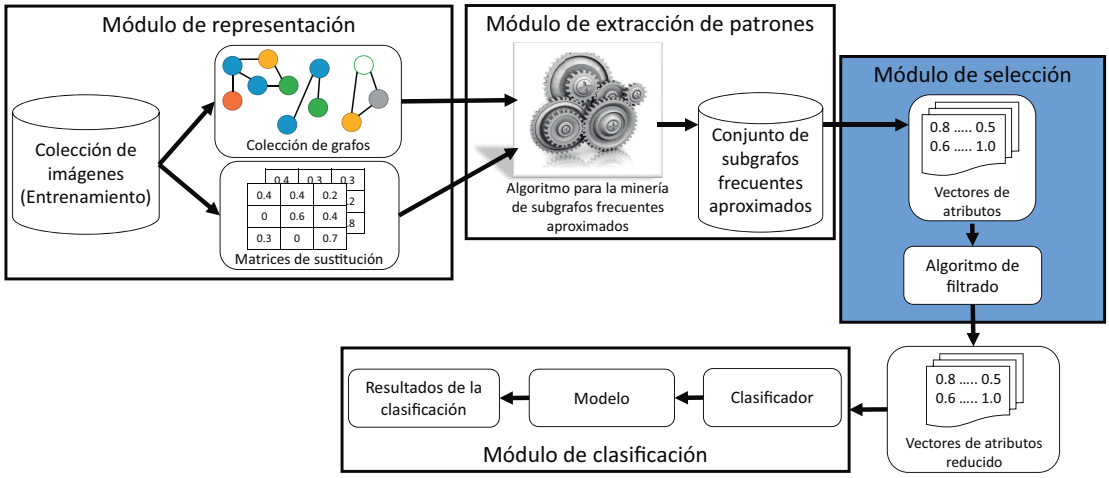

Fig. 2. Esquema de clasificación de imágenes basadas en grafos utilizando la selección de atributos.

En esta sección se realizan comparaciones entre el método de clasificación mencionado en la sección 4.2 que usa todos los SFA que fueron encontrados en la minería y el método mostrado en la figura 2, donde se selecciona un subconjunto de SFA. Lo primero que se comprara es la reducción alcanzada. En la tabla 5 se muestra la cantidad de atributos utilizados para la clasificación de ambos métodos en cada colección utilizada. Esta tabla está compuesta por cuatro conjuntos de cuatro columnas que especifican las colecciones de imágenes y una columna final que especifica un clasificador por fila. Las cuatro columnas de cada colección de imágenes indican la cantidad de atributos que se utilizaron para la clasificación con su correspondiente clasificador especificado en la última columna de la derecha de la tabla. El número de atributos seleccionados fueron obtenidos experimentalmente en los rangos: [50,300] para Coenen-DB, [50,600] para GREC, y [50,1500] para COIL.

Como se puede observar en la tabla 5, la dimensionalidad de los vectores de atributos fue reducida drásticamente utilizando los subconjuntos de atributos seleccionados mejorando la eficiencia de los algoritmos de clasificación. Esta reducción está sobre el 50 % en el 81 % de los experimentos realizados, siendo en COIL donde mayor reducción se alcanzó.

Tabla 5. Cantidad de atributos usados en el proceso de clasificación.

Coenen-DB ($\delta = 20\,\%$)				**GREC** ($\delta = 2\,\%$)				**COIL** ($\delta = 30\,\%$)				
Todos	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Classifier**
	125	275	250		425	400	525		1500	1400	1200	SVM
	100	100	100		450	500	500		1450	1500	1000	BayesNet
745	125	125	125	715	50	50	50	2668	50	50	50	AdaBoost
	250	250	300		325	275	200		1500	400	375	Reg.
	150	150	150		450	300	550		1400	50	50	D-Table.
	275	175	300		550	550	200		950	1450	550	J48graft

A continuación, en la tabla 6 se muestran los resultados de la clasificación obtenidos en las colecciones descritas en la sección 4.1 utilizando el esquema de clasificación antes presentado en la figura 2. En esta tabla se realiza una comparación entre el uso o no del módulo de selección y entre los algoritmos de selección utilizados. Esta comparación se hace con el objetivo de mostrar

la utilidad del método de clasificación basado en la MSFA que utiliza la selección de atributos. Nótese que para esta comparación se utilizan los umbrales δ que mejores resultados reportaron en la sección 4.2 para cada colección de imágenes utilizada.

Tabla 6. Resultados de la clasificación utilizando varios clasificadores sobre diferentes colecciones de grafos con y sin el uso de varios algoritmos de selección de atributos.

(a) Accuracy

		J48graft				Decision Table				Regression			
Colección	δ	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**
Coenen-DB	20 %	97.25 %	97.50 %	97.50 %	**97.75** %	94.38 %	**95.88** %	94.00 %	95.25 %	96.25 %	96.75 %	96.75 %	**96.88** %
GREC	3 %	**82.20** %	81.63 %	81.63 %	**82.20** %	65.72 %	65.72 %	**66.48** %	65.72 %	83.14 %	**85.61** %	83.52 %	82.39 %
COIL	30 %	79.96 %	**82.95** %	79.21 %	82.33 %	52.06 %	55.12 %	58.43 %	**63.17** %	74.34 %	79.56 %	81.71 %	**85.27** %
Promedio		86.47 %	87.36 %	86.11 %	**87.43** %	70.72 %	72.24 %	72.97 %	**74.71** %	84.58 %	87.31 %	87.33 %	**88.18** %

		AdaBoost				BayesNet				SVM			
Colección	δ	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**
CoenenDB	20 %	94.00 %	94.00 %	94.00 %	94.00 %	90.38 %	92.75 %	92.75 %	**93.25** %	95.38 %	95.75 %	95.75 %	**96.25** %
GREC	3 %	–	–	–	–	87.88 %	**88.07** %	87.88 %	**88.07** %	**94.51** %	92.42 %	92.61 %	93.37 %
COIL	30 %	–	–	–	–	**90.51** %	90.07 %	90.13 %	89.70 %	90.20 %	89.45 %	89.26 %	**91.14** %
Promedio		31.33 %	31.33 %	31.33 %	31.33 %	89.59 %	90.30 %	90.25 %	**90.34** %	93.36 %	92.54 %	92.54 %	**93.59** %

(b) F-measure

		J48graft				Decision Table				Regression			
Colección	δ	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**
CoenenDB	20 %	97.23 %	97.50 %	97.50 %	**97.76** %	94.49 %	**95.94** %	94.13 %	95.31 %	96.21 %	96.73 %	96.72 %	**96.86** %
GREC	3 %	**86.96** %	**86.96** %	**86.96** %	85.11 %	28.13 %	28.13 %	**30.51** %	28.13 %	78.43 %	80.00 %	85.11 %	**85.71** %
COIL	30 %	**91.18** %	82.89 %	84.56 %	**91.18** %	48.80 %	58.00 %	54.20 %	**63.60** %	67.06 %	78.08 %	**81.82** %	80.77 %
Promedio		**91.79** %	89.12 %	89.67 %	91.35 %	57.14 %	60.69 %	60.61 %	**62.35** %	80.57 %	84.94 %	**87.88** %	87.78 %

		AdaBoost				BayesNet				SVM			
Colección	δ	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**	**Todos**	**CHI-Q**	**IG**	**GRAE**
CoenenDB	20 %	93.89 %	93.89 %	93.89 %	93.89 %	90.29 %	92.37 %	92.37 %	**93.02** %	95.39 %	95.74 %	95.71 %	**96.22** %
GREC	3 %	14.50 %	14.50 %	14.50 %	14.50 %	86.96 %	86.96 %	86.96 %	86.96 %	89.36 %	88.89 %	88.89 %	**91.30** %
COIL	30 %	15.67 %	15.63 %	15.63 %	**18.31** %	**87.32** %	**87.32** %	84.35 %	85.14 %	**92.19** %	90.37 %	86.52 %	89.71 %
Promedio		41.31 %	41.31 %	41.31 %	**42.23** %	88.19 %	**88.88** %	87.89 %	88.37 %	92.31 %	91.67 %	90.37 %	**92.41** %

La tabla 6 está compuesta por dos subtablas que muestran los resultados del (a) accuracy y (b) F-measure respectivamente. La primera y segunda columna de estas subtablas especifican la colección utilizada y el valor del umbral de mínimo soporte respectivamente. Luego, las subtablas se dividen en tres conjuntos de columnas que representan los resultados alcanzados con los clasificadores especificados en la parte superior de estas. Cada conjunto de columnas está compuesto por cuatro columnas que indican los resultados de la clasificación utilizando: todos los atributos, los atributos seleccionados con CHI-Q, los seleccionados con IG, y los seleccionados con GRAE, respectivamente. Finalmente se muestran los promedios de los resultados de varios clasificadores.

Como se puede observar en la tabla 6, los resultados de la clasificación alcanzados utilizando el método de clasificación mencionado en esta sección son competitivos con los resultados logrados con el método mencionado en la sección 4.2, y es importante señalar que al utilizar el módulo de selección se utiliza un número mucho menor de atributos.

Adicionalmente, en la tabla 7 se presentan comparaciones de significancia estadística entre los clasificadores utilizando todos los atributos siguiendo el método de la sección 4.2 y utilizando solo los atributos seleccionados por los diferentes algoritmos de selección del método propuesto en esta sección. Para esta comparación se utiliza una prueba de significancia estadística reportada por García y Herrera [40] (*Bergmann* [41]) con 0.5 y 0.10 como valores de α. En la primera columna

de la tabla 7, "Todos" representa el método que utiliza todos los atributos calculados por VEAM mientras "IG", "CHI-Q" y "GRAE" representan el método que incluye el módulo de selección mediante ganancia de información, chi-cuadrado y cociente de evaluación de la ganancia de información, respectivamente. Las columnas restantes muestran el enfoque que se identificó como mejor opción según la prueba de significancia estadística. El símbolo "–" indica que no existe diferencias estadísticamente significativas entre los diferentes enfoques.

Como se puede observar en las tablas 5, 7 y 6, el uso de algoritmos de selección de atributos es de ayuda para el desarrollo de un mejor método de clasificación de imágenes basado en la MSFA. El algoritmo de selección GRAE es la mejor opción entre los selectores dado los resultados globales que se mostraron en las tablas mencionadas, donde se logra una considerable reducción de atributos y una buena calidad en los resultados de la clasificación.

Tabla 7. Pruebas de significancia estadística para diferentes clasificadores en varias colecciones de grafos (imágenes) utilizando todos los atributos y usando los atributos seleccionados por diferentes algoritmos de selección de atributos, donde α =0.05 y α =0.10.

Classifier	J48graft	Decision Table	Regression	AdaBoost	BayesNet	SVM
Todos vs. GRAE	–	–	GRAE	–	–	–
Todos vs. CHI-Q	–	–	–	–	–	–
Todos vs. IG	–	–	IG	–	–	–
IG vs. GRAE	–	–	–	–	–	–
CHI-Q vs. GRAE	–	–	–	–	–	–
IG vs. CHI-Q	–	–	–	–	–	–

4.4. Mejorando en eficiencia

La MSFA tiene mayor complejidad computacional que los métodos exactos debido al cálculo de las semejanzas entre subestructuras y el número de candidatos que esto trae consigo es mayor, así como el número de candidatos duplicados. Un candidato duplicado es un subgrafo que fue considerado en pasos previos, pero aparece nuevamente a partir de varios subgrafos frecuentes durante la búsqueda. La aparición de estos duplicados en el proceso de minería es uno de los mayores problemas a enfrentar tanto en los métodos exactos como en los aproximados. Este problema se trata realizando pruebas de formas canónicas para representar los subgrafos como un código único (forma canónica) y mediante comparaciones entre estas formas canónicas se detectan dichos duplicados. Sin embargo, estas pruebas de formas canónicas tienen una gran complejidad computacional y aumenta esta complejidad mientras mayor sea el subgrafo. Por lo que para disminuir la cantidad de duplicados se han desarrollado varias podas que aceleran el proceso de la minería [3,21,26,27]. Sin embargo, solo algunas han sido aplicadas a la MSFA [21,26,27] logrando una reducción considerable del espacio de búsqueda de las etiquetas y el número de pruebas de formas canónicas basándose en la propiedad de clausura-descendente, como los algoritmos APGM y VEAM. La base de dichas podas reside en el procesamiento de la información existente en las matrices de sustitución que utilizan estos algoritmos (ver sección 3.2).

En la tabla 8 se muestran algunos de los resultados alcanzados al introducir las podas propuestas [26,27] para el proceso de MSFA de VEAM. Primero se compara el algoritmo original de VEAM con él mismo haciendo uso de dichas podas, denotado por VEAMwP, según la cantidad

Tabla 8. Comparación entre VEAM and VEAMwP en varias colecciones de grafos (imágenes) utilizando diferentes umbrales de soporte mínimo δ.

(a) Cantidad de pruebas de formas canónicas realizadas

		Soporte (δ)								
Colección	**Algoritmo**	20 %	25 %	30 %	35 %	40 %	45 %	50 %	55 %	60 %
CoenenDB	VEAM	316120	130396	56765	12176	4627	2709	2250	2127	1686
	VEAMwP	**221060**	**77406**	**24710**	**5409**	**1999**	**466**	**426**	**400**	**376**
		2 %	3 %	4 %	5 %	6 %	7 %	8 %	9 %	10 %
GREC	VEAM	135284	73088	45958	32142	22867	17358	13471	11366	9865
	VEAMwP	**127895**	**69241**	**43286**	**29983**	**21448**	**16376**	**12489**	**10295**	**8980**
		30 %		40 %	50 %	60 %	70 %	80 %	90 %	100 %
COIL	VEAM	7151952		784643	138962	81554	38515	29334	0	0
	VEAMwP	**6726947**		**742939**	**133906**	**78497**	**37091**	**28247**	0	0

(b) Cantidad de duplicados identificados

		Soporte (δ)								
Colección	**Algoritmo**	20 %	25 %	30 %	35 %	40 %	45 %	50 %	55 %	60 %
CoenenDB	VEAM	1762	569	202	23	5	1	0	0	0
	VEAMwP	**1312**	**518**	**106**	**16**	5	1	0	0	0
		2 %	3 %	4 %	5 %	6 %	7 %	8 %	9 %	10 %
GREC	VEAM	1352	627	340	218	136	91	58	48	39
	VEAMwP	**1163**	**524**	**298**	**192**	**115**	**81**	**51**	**41**	**33**
		30 %		40 %	50 %	60 %	70 %	80 %	90 %	100 %
COIL	VEAM	27		0	0	0	0	0	0	0
	VEAMwP	**15**		0	0	0	0	0	0	0

(c) Tiempo de ejecución (s)

		Soporte (δ)								
Colección	**Algoritmo**	20 %	25 %	30 %	35 %	40 %	45 %	50 %	55 %	60 %
CoenenDB	VEAM	122.37	48.62	22.98	6.74	3.52	2.57	2.31	2.17	1.97
	VEAMwP	**66.70**	**23.16**	**9.34**	**3.49**	**2.31**	**0.92**	**0.87**	**0.83**	**0.81**
		2 %	3 %	4 %	5 %	6 %	7 %	8 %	9 %	10 %
GREC	VEAM	1.19	0.81	0.64	0.52	0.45	0.41	0.36	0.33	0.31
	VEAMwP	**0.95**	**0.59**	**0.46**	**0.40**	**0.33**	**0.27**	**0.23**	**0.21**	**0.20**
		30 %		40 %	50 %	60 %	70 %	80 %	90 %	100 %
COIL	VEAM	1337.61		191.00	31.32	21.60	11.65	9.60	0	0
	VEAMwP	**829.68**		**130.83**	**29.80**	**20.70**	**11.13**	**9.44**	0	0

exhaustiva de pruebas de formas canónicas que estos realizan en el proceso de MSFA (ver la sub-tabla (a) de la tabla 8). Es importante señalar que en esta comparación se logra una reducción mayor de 35 % de la cantidad de pruebas de formas canónicas cuando se utilizan las podas. La cantidad de duplicados identificados también disminuye considerablemente permitiendo una mejora en el procesamiento de los datos. Además, se realiza una comparación entre VEAM y VEAMwP en términos de tiempo de ejecución (ver sub-tabla (b) de la tabla 8). En esta última comparación se logra una reducción mayor de 45 % de los tiempos de ejecución. Los resultados mostrados en la tabla 8 son alcanzados sobre el conjunto de entrenamiento de cada colección descrita en la sección 4.1.

Como se puede observar en la tabla 8, la reducción del espacio de búsqueda de las etiquetas y la cantidad de candidatos a procesar permite mejorar el comportamiento de VEAM. Estos resultados permiten afirmar que estas podas son útiles para los procesos de MSFA similares al de VEAM.

4.5. Síntesis y conclusiones

Como se muestra a lo largo de esta sección, muchos han sido los esfuerzos orientados al mejoramiento de la clasificación tanto en eficiencia como en eficacia de los métodos basados en la MSFA. Todos los trabajos mencionados han tenido un impacto faborable en las tareas de clasificación de imágenes. Con los aportes realizados por varios autores se ha logrado la disminución tanto del tiempo de procesamiento en la MSFA como de la dimensionalidad de los patrones a utilizar en la clasificación con mejoras relevantes en sus resultados.

5. Aplicaciones de la MSFA

Los grafos son estructuras de datos generales y con un gran poder descriptivo con los que se pueden modelar de forma natural objetos y datos de múltiples dominios de la ciencia. En estos dominios, donde los objetos son representables en forma de grafos y existe una cantidad necesaria de datos para realizar la búsqueda de conocimiento, es donde se puede aplicar la minería de subgrafos frecuentes (MSF). Como se ha mencionado anteriormente, cuando los objetos semejantes modelados en forma de grafos presentan algunas variaciones que no le permiten ser idénticos, algo muy común en problemas prácticos, entonces es necesaria la aplicación de la MSFA. La utilización de este tipo de minería es más efectiva donde las variaciones en los vértices y aristas contengan un valor semántico interesante para usuarios. La MSFA es efectiva en aquellos tipos de datos que pueden ser modelados semánticamente a través de Marcos o Mapas conceptuales [42], Ontologías [43], Redes semánticas [44], entre otros, donde se puede realizar un proceso de MSFA para identificar nuevos conceptos aproximados que aporten nuevo conocimiento en aplicaciones concretas.

Existen numerosas aplicaciones basadas en los tipos de representaciones antes mencionadas que han utilizado procesos de minería; por ejemplo: clasificación de documentos [45]; análisis de comunidades Web, extracción automática de tópicos desde documentos Web y clasificación de páginas Web mediante su estructura [46]. Sin embargo, no tienen en cuenta las aproximaciones semánticas en la minería. Varias de las aplicaciones basadas en la representación de los datos mediante Marcos conceptuales u Ontologías, en las que se puede utilizar la MSFA, pudieran ser: clasificación de documentos, agrupamiento de documentos por contenido e identificación de tópicos [47], detección de correos sospechosos y clasificación de tópicos [48]. Otras posibles aplicaciones de la MSFA, que más se ajustan a los problemas prácticos y que pudieran utilizar las Redes semánticas como forma de representación, son la detección de tendencias, comportamientos y regularidades sobre Redes Sociales [49]. Estas redes sociales también pueden ser representadas mediante una ontologías de tipo DRO (de sus siglas en inglés, *Domain Reference Ontology*) [50], la cual representa la naturaleza semántica que subyace en los datos que se representan en ella ofreciendo un mayor nivel de expresividad respecto a los demás tipos existentes.

Por otro lado, las estructuras o modelos de las bases de datos no son consideradas como una posible aplicación de la MSFA donde se contemplen aproximaciones en las aristas. Estos modelos están compuestos por entidades o tablas (vértices) y las relaciones entre ellas (aristas), lo que se conoce como modelos relacionales. Las relaciones entre las entidades solo son de tres tipos: relación de mucho a mucho, relación de uno a mucho y relación de uno a uno. Como se puede deducir, estas relaciones no poseen información semántica interesante para el proceso de minería aproximado completo. Este mismo efecto puede verse reflejado en el análisis de estructuras químicas, donde

también existen a lo sumo 3 tipos de relaciones entre los vértices por lo que las semejanzas entre estas es muy reducida. Por esta razón, se ha llegado a la conclusión de que en estas dos áreas no es posible una aplicación satisfactoria de la MSFA tratada en este trabajo, ya que el tipo de minería que se analiza en este trabajo incluye las aproximaciones entre las etiquetas de las aristas.

6. Conclusiones

En este trabajo se realiza un estudio de los aportes reportados en la literatura donde se utiliza la MSFA, específicamente los que permiten distorsiones semántica de las etiquetas manteniendo la topología de los grafos, en tareas de clasificación de imágenes. Los resultados mostrados en este trabajo ilustran la importancia y la utilizad del uso de la MSFA en tareas de clasificación de imágenes. Con el uso de la MSFA se obtuvieron buenos resultados en esta área con algunas mejoras en la eficacia mediante el uso de filtros o algoritmos de selección de atributos. El uso de estos algoritmos de selección no solo permitió una mejora en la eficacia del método de clasificación sino que se logró una considerable disminución de la dimensionalidad de los vectores de atributo, lo cual repercutió en un aumento de la eficiencia del método. Además, se mostraron algunas podas reportadas para la MSFA que permitieron una ganancia en eficiencia logrando una reducción del espacio de búsqueda de las etiquetas y una disminución de la cantidad de pruebas de formas canónicas a realizar. Con estas podas se trata uno de los problemas medulares de este tipo de minería: el problema de la eficiencia. Finalmente, partiendo de un estudio realizado se identificaron varias áreas de aplicación con posibles resultados de impacto en diferentes dominos de la ciencia.

Referencias bibliográficas

1. Eichinger, F., Böhm, K.: Software-Bug Localization with Graph Mining. In Aggarwal, C.C., Wang, H., eds.: Managing and Mining Graph Data. Volume 40 of Advances in Database Systems. Springer-Verlag New York (2010)
2. Yan, X., Huan, J.: gspan: Graph-based substructure pattern mining. In: Proc. Int'l Conf. Data Mining. (2002)
3. Gago-Alonso, A., Carrasco-Ochoa, J.A., Medina-Pagola, J.E., Martínez-Trinidad, J.F.: Full Duplicate Candidate Pruning for Frequent Connected Subgraph Mining. Integrated Computer-Aided Engineering **17** (August 2010) 211–225
4. Gago-Alonso, A., Puentes-Luberta, A., Carrasco-Ochoa, J.A., Medina-Pagola, J.E., Martínez-Trinidad, J.F.: A new algorithm for mining frequent connected subgraphs based on adjacency matrices. Intelligent Data Analysis **14** (2010) 385–403
5. Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2004) 647–652
6. Holder, L.B., Cook, D.J., Bunke, H.: Fuzzy substructure discovery. In: ML92: Proceedings of the ninth international workshop on Machine learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1992) 218–223
7. Ketkar, N., Holder, L., Cook, D.: Mining in the proximity of subgraphs. Analysis and Group Detection KDD Workshop on Link Analysis: Dynamics and Statics of Large Networks (August 2006)
8. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: Proceedings of the 3rd IEEE International Conference on Data Mining. (2003) 549–552
9. Hossain, M.S., Angryk, R.A.: Gdclust: A graph-based document clustering technique. In: ICDMW '07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, Washington, DC, USA, IEEE Computer Society (2007) 417–422
10. Borgelt, C.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proc. of 2002 IEEE International Conference on Data Mining (ICDM, IEEE Press (2002) 51–58

11. Jiang, C., Coenen, F.: Graph-based Image Classification by Weighting Scheme. In: Proceedings of the Artificial Intelligence, Springer, Heidelberg (2008) 63–76
12. Elsayed, A., Coenen, F., Jiang, C., nana, F.G.F., Sluming, V.: Corpus Callosum MR Image Classification. Knowledge-Based Systems **23** (2010) 330–336
13. Fellman, P.V.: Modeling terrorist networks-complex systems at the mid-range. In: Downloaded from the internet. (November 2008)
14. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI **18**(3) (2004) 265–298
15. Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.E.: Frequent approximate subgraphs as features for graph-based image classification. Knowledge-Based Systems **27** (2012) 381–392
16. Acosta-Mendoza, N., Morales-González, A., Gago-Alonso, A., García-Reyes, E.B., Medina-Pagola, J.E.: Classification using Frequent Approximate Subgraphs. In: Proceedings of the 17th Iberoamerican Congress on Pattern Recognition (CIARP'12). Volume 7441., Buenos Aires, Argentina, Springer-Verlag Berlin Heidelberg (September 2012) 292–299
17. Acosta-Mendoza, N., Gago-Alonso, A., Medina-Pagola, J.E.: Clasificaión de imágenes utilizando minería de subgrafos frecuentes aproximados. Revista Cubana de Ciencias Informáticas (RCCI) **5**(4) (2012) 1–10
18. Morales-González, A., Acosta-Mendoza, N., Gago-Alonso, A., García-Reyes, E.B., Medina-Pagola, J.E.: A new proposal for graph-based image classification using frequent approximate subgraphs. Pattern Recognition (0) (2013) –
19. Acosta-Mendoza, N., Gago-Alonso, A., Carrasco-Ochoa, J.A., MArtínez-Trinidad, J.F., Medina-Pagola, J.E.: Feature Space Reduction for Graph-Based Image Classification. In: Proceedings of the 18th Iberoamerican Congress on Pattern Recognition (CIARP'13), Havana, Cuba, Springer-Verlag Berlin Heidelberg (november 2013) –
20. Jia, Y., Huan, J., Buhr, V., Zhang, J., Carayannopoulos, L.: Towards comprehensive structural motif mining for better fold annotation in the "twilight zone" of sequence dissimilarity. BMC Bioinformatics **10**(S-1) (2009)
21. Jia, Y., Zhang, J., Huan, J.: An efficient graph-mining method for complicated and noisy data with real-world applications. Knowledge Information Systems **28**(2) (2011) 423–447
22. Chen, C., Yan, X., Zhu, F., Han, J.: gapprox: Mining frequent approximate patterns from a massive network. In: International Conference on Data Mining (ICDM'07). (2007) 445–450
23. Xiao, Y., Wang, W., Wu, W.: Mining conserved topological structures from large protein-protein interaction networks. In: Proceedings of the 18th IEICE data engineering workshop / 5th DBSJ annual meeting, Hiroshima, Japan, DEWS'2007 (2007)
24. Song, Y., Chen, S.S.: Item sets based graph mining algorithm and application in genetic regulatory networks. Data Mining, IEEE International Conference on Volume, Issue (2006) 337–340
25. Zhang, S., Yang, J., Cheedella, V.: Monkey: Approximate graph mining based on spanning trees. In: International Conference on Data Engineering, Los Alamitos, CA, USA, IEEE ICDE (2007) 1247–1249
26. Acosta-Mendoza, N., Alonso, A.G., Medina-Pagola, J.E.: Mejora para la minería de subgrafos frecuentes aproximados mediante la reducción del espacio de búsqueda. In: IX Congreso Nacional de Reconocimiento de Patrones (RECPAT), Santa Clara, Cuba (noviembre 2011)
27. Acosta-Mendoza, N., Alonso, A.G., Medina-Pagola, J.E.: On Speeding up Frequent Approximate Subgraph Mining. In: Proceedings of the 17th Iberoamerican Congress on Pattern Recognition (CIARP'12). Volume 7441., Buenos Aires, Argentina, Springer-Verlag Berlin Heidelberg (september 2012) 316–323
28. Zhang, S., Yang, J.: Ram: Randomized approximate graph mining. In: Proceedings of the 20th International Conference on Scientific and Statistical Database Management, Hong Kong, China (2008) 187–203
29. Xiao, Y., Wu, W., Wang, W., He, Z.: Efficient algorithms for node disjoint subgraph homeomorphism determination. In: Proceedings of the 13th International Conference on Database Systems for Advanced Applications, New Delhi, India (2008) 452–460
30. Zou, Z., Li, J., Gao, H., Zhang, S.: Frequent subgraph pattern mining on uncertain graph data. In: CIKM'09: Proceeding of the 18th ACM conference on Information and knowledge management, New York, NY, USA, ACM (2009) 583–592
31. Zou, Z., Li, J., Gao, H., Zhang, S.: Mining frequent subgraph patterns from uncertain graph data. IEEE Trans. on Knowl. and Data Eng. **22**(9) (2010) 1203–1218
32. Papapetrou, O., Ioanno, E., Skoutas, D.: Efficient discovery of frequent subgraph patterns in uncertain graph databases. In: Proceedings of the 14th International Conference on Extending Database Technology, New York, NY, USA (2011) 355–366
33. Acosta-Mendoza, N.: Clasificación de imágenes basada en subconjunto de subgrafos frecuentes aproximados. Master's thesis, The National Institute of Astrophysics, Optics and Electronics of Mexico (INAOE) (July 2013)

34. S.Nene, S.Nayar, H.Murase: Columbia object image library (coil-100). Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshop, SSPR & SPR 2008 (2008)
35. Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning, Orlando, USA (2008) 208–297
36. Brun, L., Kropatsch, W.: Introduction to combinatorial pyramids. Digital and image geometry: advanced lectures (2001) 108–128
37. Kropatsch, W., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision pyramids that do not grow too high. Pattern Recognition Letters **26(3)** (2005) 319–337
38. Morales-González, A., García-Reyes, E.B.: Simple object recognition based on spatial relations and visual features represented using irregular pyramids. Multimedia Tools and Applications (2011) 1–23
39. Finkel, R.A., Bentley, J.L.: Quad trees: A data structure for retrieval on composite keys. Acta Informatica **4** (1974) 1–9
40. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. Journal of Machine Learning Research (2008) 2677–2694
41. Bergmann, G., Hommel, G.: Improvements of general multiple test procedures for redundant systems of hypotheses. In: P.Bauer, G. Hommel, and E. Sonnemann, editors, Multiple Hypotheses Testing, Springer, Berlin (1988) 100–115
42. Minsky, M.: A framework for representing knowledge. In Haugeland, J., ed.: Mind Design: Philosophy, Psychology, Artificial Intelligence. MIT Press, Cambridge, MA (1981) 95–128
43. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2) (june 1993) 199–220
44. Lehmann, F.: Semantic Networks in Artificial Intelligence. Elsevier Science Inc., New York, NY, USA (1992)
45. Jiang, C., Coenen, F., Sanderson, R., Zito, M.: Text classification using graph mining-based feature extraction. Knowledge-Based Systems **23**(4) (2010) 302–308
46. Xu, G., Zhang, Y., Li, L.: Web Mining and Social Networking: Techniques and Applications. Web Information Systems Engineering and Internet Technologies Book Series. Springer (2010)
47. Pérez-Suárez, A., Martínez-Trinidad, J., Carrasco-Ochoa, J., Medina-Pagola, J.: A dynamic clustering algorithm for building overlapped clusters. To appear in Journal Intelligent Data Analysis **16(2)** (2012)
48. Nizamani, S., Memon, N., Wiil, U.K., Karampelas, P.: Ccm: A text classification model by clustering. In: International Conference on Advances in Social Networks Analysis and Mining (ASONAM'11), Kaohsiung, Taiwan, IEEE Computer Society (2011) 461–467
49. Memon, N., Xu, J.J., Hicks, D.L., Chen, H., eds.: Data Mining for Social Network Data. Volume 12 of Annals of Information Systems. Springer (2010)
50. Fonseca, R.L., Llano, E.G.: Automatic representation of geographical data from a semantic point of view through a new ontology and classification techniques. Transaction in GIS **15**(1) (2011) 61–85

RT_027, junio 2014
Aprobado por el Consejo Científico CENATAV

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