
Descubrimiento de Conocimiento Basado en Grafos

Knowledge Discovery Based on Graphs

Augusto Cortez Vásquez, Luzmila Pro Concepción

Universidad Nacional Mayor de San Marcos

Facultad de Ingeniería de Sistemas e Informática
cortez_augusto@yahoo.fr, Lproc2003@hotmail.com,

RESUMEN

De la misma forma que las bases de datos relacionales organizan los datos en forma de tablas, las bases de datos de grafos lo hacen en forma de grafos. Una de las principales razones que nos puede llevar a utilizar una base de datos basada en grafos BDBG, es que la importancia reside ampliamente en las relaciones entre entidades, como por ejemplo aplicaciones de moda tales como redes sociales, minería de datos, entre otras. Todas éstas cumplen este tipo de condiciones. Ahora bien, una característica muy importante de las BDBG, es que la información más importante no se encuentra en los nodos, sino en las relaciones entre estos mismos, es importante sobretodo en las redes sociales en la que se complica la cosa cuando empezamos a ver relaciones no solamente entre las personas sino entre sus actividades. De ahí la diferencia con las bases de datos relacionales y de objetos, las cuales dan prioridad a los datos, y las relaciones son una forma de navegar entre ellos. Por otra parte, con un contenido basado en el usuario, es muy difícil preconcebir el esquema exacto de los datos que se manejarán. Por esto las BD relacionales, necesitan esquemas iniciales no dinámicos, mientras que las BDBG pueden soportar el dinamismo de la información. Si el objetivo principal es analizar las relaciones entre las entidades, el presente artículo intenta persuadir al lector de la fuerza de la evidencia que sustenta el punto de vista que se trata de exponer.

Palabras claves: Conocimiento Basado en Grafos, base de datos basada en grafos., extracción de conocimiento basado en grafos.

ABSTRACT

In the same way as relational databases organize data in the form of tables, graph databases do in the form of graphs. One of the main reasons that could lead us to use a database based on graph BDBG, is that the importance lies largely in relations between entities, such as for example applications of fashion such as social networks, mining data, among others. All of these meet this kind of conditions. Now, a very important feature of the BDBG, is that the most important information is not in the nodes, but in relations between these same is important especially in social networks in which the thing is complicated when we begin to see relations not only between individuals but between their activities. Hence the difference with relational databases and objects, which give priority to data, and relationships are a way to navigate between them. On the other hand, with a content based on the user, it is very difficult to preconcebir the exact schema of data to handle. Therefore the relational bond, they need not dynamic initial schemes, while the BDBG can withstand the dynamism of the information. If the main objective is to analyze the relationships between the entities, it will try to persuade the reader of the strength of the evidence that supports the point of view I try to expose.

Key Words: Knowledge based on graph, database-based graph., graph-based knowledge extraction.

1. INTRODUCCION

Un área que ha mostrado gran interés en los últimos años es la que se conoce como sistema de descubrimiento de conocimiento basado en grafos. Ahora bien, la cuestión que debemos considerar es que estos sistemas utiliza un algoritmo de minería de datos que representa el conocimiento mediante grafos. Para ello la fase de preparación de datos incluye una transformación de los datos a un formato de grafo .

el modelo de representación que utiliza subdue es un grafo etiquetado. Los objetos se representan con vértices y la relaciones con arcos. Las etiquetas se utilizan para describir el significado de los arcos y vértices.

2. FUNDAMENTACION TEORICA

2.1 Grafos

Grafo

Un grafo es una estructura de datos que consiste en un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto. Los nodos representan conceptos y los arcos representan relaciones entre ellos.

en los sistemas actuales, los nodos pueden representar estructuras arbitrariamente complejas (ejm. marcos), mientras que las relaciones plasmados mediante los arcos pueden pertenecer a diversas clases diferenciados (ejm. causales, temporales, de atributo, etc.)

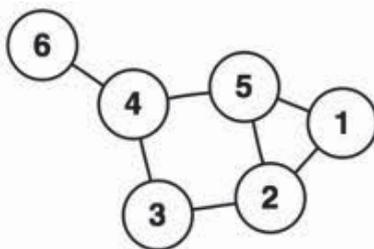


Figura 1. Grafo

Típicamente, un grafo se representa gráficamente como un conjunto de puntos (vértices o nodos) unidos por líneas (aristas).

Desde un punto de vista práctico, los grafos permiten estudiar las interrelaciones entre unidades que inte-

ractúan unas con otras. Por ejemplo, una red de computadoras puede representarse y estudiarse mediante un grafo, en el cual los vértices representan terminales y las aristas representan conexiones (las cuales, a su vez, pueden ser cables o conexiones inalámbricas).

Prácticamente cualquier problema puede representarse mediante un grafo, y su estudio trasciende a las diversas áreas de las ciencias exactas y las ciencias sociales.

REPRESENTACIÓN DE GRAFOS

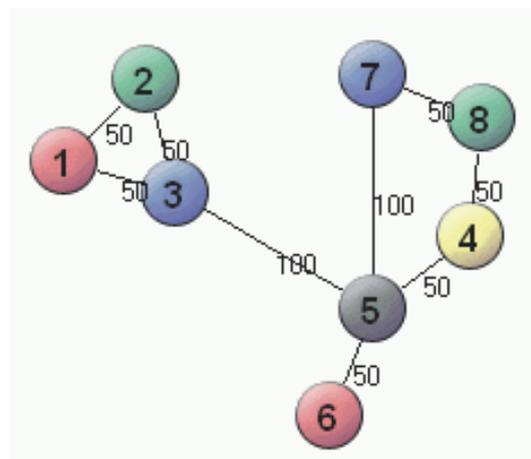


Figura 2. grafo con pesos.

Hay tres maneras de representar un grafo en un programa: mediante matrices, mediante listas y mediante matrices dispersas.

- **Representación mediante matrices:** La forma más fácil de guardar la información de los nodos es mediante la utilización de un vector que indexe los nodos, de manera que los arcos entre los nodos se pueden ver como relaciones entre los índices. esta relación entre índices se puede guardar en una matriz, que llamaremos de adyacencia.
- **Representación mediante listas:** en las listas de adyacencia lo que haremos será guardar por cada nodo, además de la información que pueda contener el propio nodo, una lista dinámica con los nodos a los que se puede acceder desde él. La información de los nodos se puede guardar en un vector, al igual que antes, o en otra lista dinámica.
- **Representación mediante matrices dispersas:** Para evitar uno de los problemas que teníamos con

las listas de adyacencia, que era la dificultad de obtener las relaciones inversas, podemos utilizar las matrices dispersas, que contienen tanta información como las matrices de adyacencia, pero, en principio, no ocupan tanta memoria como las matrices, ya que al igual que en las listas de adyacencia, sólo representaremos aquellos enlaces que existen en el grafo.

Recorrido de un grafo

Recorrer un grafo significa tratar de alcanzar todos los nodos que estén relacionados con uno que llamaremos nodo de salida. existen básicamente dos técnicas para recorrer un grafo: el recorrido en anchura; y el recorrido en profundidad.

- **Recorrido en anchura:** el recorrido en anchura supone recorrer el grafo, a partir de un nodo dado, en niveles, es decir, primero los que están a una distancia de un arco del nodo de salida, después los que están a dos arcos de distancia, y así suce-

sivamente hasta alcanzar todos los nodos a los que se pudiese llegar desde el nodo salida. el tiempo de ejecución como el espacio de memoria necesario, crece en de forma exponencial con el tamaño del problema.

- **Recorrido en profundidad:** el recorrido en profundidad trata de buscar los caminos que parten desde el nodo de salida hasta que ya no es posible avanzar más. Cuando ya no puede avanzarse más sobre el camino elegido, se vuelve atrás en busca de caminos alternativos, que no se estudiaron previamente.

Recorrido en anchura

se recorren de izquierda a derecha, todos los subobjetivos generados por el objetivo de nivel inmediatamente superior antes de considerar el siguiente nivel de profundidad del árbol (Cuadro 1).

es un procedimiento completo: garantiza encontrar una solución si esta existe, aunque no sea la solución óptima. La solución sería JAGHBCIEFDK

Cuadro 1

J											
	A	G	H								Extraer J + hijos de J
		G	H	B	C						Extraer A + hijos de A
			H	B	C	I					Extraer G+ hijos de G
				B	C	I	E	F			Extraer H + hijos de H
					C	I	E	F	D	K	Extraer B + hijos de B
						I	E	F	D	K	Extraer C
							E	F	D	K	Extraer I

Cuadro 2

J											
	A	G	H								Extraer J + hijos de J
		B	C	G	H						Extraer A + hijos de A
			D	K	C	G	H				Extraer B+ hijos de B
				K	C	G	H				Extraer D
					C	G	H				Extraer K
						G	H				Extraer C
							I	H			Extraer G+ hijos de G
								H			Extraer I
									E	F	Extraer H+ hijos de H

Recorrido en profundidad

Se recorren de izquierda a derecha, todos los subobjetivos generados por el objetivo de nivel inmediatamente superior antes de considerar el siguiente nivel de profundidad del árbol (Cuadro 2).

es un procedimiento completo: garantiza encontrar una solución si esta existe, aunque no sea la solución óptima. La solución sería JABDKCGIHEF

La complejidad espacial se reduce respecto a la anterior ya que solo es necesario guardar constancia del camino construido hasta el momento. La complejidad para un camino de longitud p será dicho valor mas el factor de ramificación r , es decir $O(p+r)$

Recorrido por primero el mejor

La búsqueda por primero el mejor () es una estrategia de control sistemática que combina las ventajas de los métodos de búsqueda exhaustiva en amplitud y en profundidad. su principal diferencia con estos métodos es que en este caso se hace uso de la información que proviene de las funciones de evaluación para ordenar los nodos en la cola.

2.2 Conocimiento

La historia del concepto de conocimiento comenzó con la formulación que hicieron los griegos ¿Qué es la realidad?, algunos autores han llegado a la conclusión de que la teoría del conocimiento es la disciplina filosófica más importante. En muchas áreas del saber, el conocimiento se ha venido obteniendo por el clásico

método hipotético-deductivo de la ciencia positiva. en él es fundamental el paso inductivo inicial: a partir de un conjunto de observaciones y de unos conocimientos previos, la intuición del investigador le conduce a formular la hipótesis. esta "intuición" resulta inoperante cuando no se trata de observaciones aisladas y casuales, sino de millones de datos almacenados en soporte informático. en el fondo de todas las investigaciones sobre inducción en bases de datos subyace la idea de automatizar ese paso inductivo.

Sierra Diez (Representacion del conocimiento, colección estructuras y procesos 1994) señala que todo el conocimiento humano estaría organizado en esquemas. Para autores como Norman, Rumelhart y otros los esquemas estarían en otros formando estructuras recursivos.

Las técnicas de análisis estadístico, desarrolladas hace tiempo, permiten obtener ciertas informaciones útiles, pero no inducir relaciones cualitativas generales, o leyes, previamente desconocidas; para esto se requieren técnicas de análisis inteligente ([Frawley et al., 91]) que todavía no han sido perfectamente establecidas. Por ello, se incrementa de forma continua la diferencia existente entre la cantidad de datos disponibles y el conocimiento extraído de los mismos.

Base de conocimiento

Generalmente una base de conocimiento es un depósito de la información que poseemos sobre unos objetos y sus relaciones dentro de un dominio específico del mundo real. el conocimiento obtenido en una base de

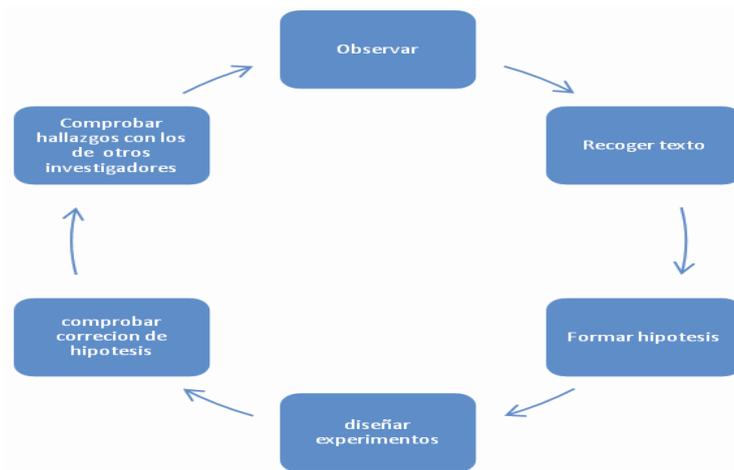


Figura 3. Pasos para descubrimiento de conocimiento

conocimiento será generalmente de distintos tipos, y admitirá por lo tanto diversas formalizaciones para su representación y mantenimiento[3]

Pero cada vez más investigaciones dentro de la inteligencia artificial están enfocadas a la inducción de conocimiento en bases de datos. Consecuencia de esta creciente necesidad ha aparecido un nuevo campo de interés: la minería de datos (data mining), que incluye los nuevos métodos matemáticos y técnicas software para análisis inteligente de datos. La minería de datos surge a partir de sistemas de aprendizaje inductivo en computadores, al ser aplicados a bases de datos ([Holshemer y Siebes, 94]), y su importancia crece de tal forma que incluso es posible que, en el futuro, los sistemas de aprendizaje se usen de forma masiva como herramientas para analizar datos a gran escala.

Descubrimiento de conocimiento DC

- Walker señala que el descubrimiento consiste de los siguientes pasos:

Durante las últimas décadas, la tecnología de las computadoras electrónicas ha hecho enormes progresos. y estoy seguro de que en las próximas décadas tendrán lugar nuevos progresos en velocidad, capacidad y diseño lógico, que permitirán el incremento de su potencial de utilización para realizar el paso 1 y 2, los programas estadísticos pueden generar agrupaciones de forma automática entre los datos recogidos, aunque no siempre se corresponden con las clasificaciones hechas por los hombres; también hay programas con cierta capacidad para diseñar experimentos; y algunos sistemas robóticos realizan las manipulaciones necesarias en ciertos experimentos.

se han levantado diversas objeciones contra el argumento de la capacidad de los computadores, existen, sin embargo según algunos teóricos la posición de que los computadores no tienen la habilidad ni capacidad para adaptarse a nuevas situaciones o aplicarlas a nuevos problemas; en este sentido, los computadores no tienen capacidad de descubrir.

en realidad, las actividad de DC no requiere la realización simultánea de todas estas tareas. Un investigador puede descubrir nuevo conocimiento a través del análisis de sus datos, mientras que un computador puede examinar los datos disponibles o recogidos por otros computadores y encontrar relaciones y explicaciones previamente desconocidas, realizando

así descubrimiento en un sentido más restringido. La capacidad de los computadores para realizar búsquedas exhaustivas de forma incansable entre grandes cantidades de datos ofrece buenas expectativas para obtener descubrimiento de forma automática [20].

- Frawley por su parte, en su exposición, parece aceptar implícitamente que el DC consiste en la extracción no trivial de información implícita, previamente desconocida y potencialmente útil, a partir de un conjunto de datos. Dado un conjunto de hechos (datos) H, un lenguaje L, y alguna medida de la certidumbre C, definimos una regularidad (pattern) como una sentencia S en L que describe relaciones dentro de un subconjunto Hs de H con una certidumbre c, de forma que s es más sencillo que la enumeración de todos los hechos de Hs. Una regularidad que sea interesante y bastante cierta (según criterios definidos por el usuario) se denomina conocimiento. Un sistema de descubrimiento será un programa que toma como entrada el conjunto de hechos y extrae las regularidades existentes. Cuando el conocimiento se extrae partiendo de los datos de una base de datos, se tiene KDD.

Descubrimiento de Conocimiento en Bases de Datos (KDD)

en un sentido muy amplio se considera descubrimiento de conocimiento en bases de datos (KDD) al proceso global de búsqueda de nuevo conocimiento a partir de los datos de una base de datos. este proceso incluye no sólo el análisis inteligente de los datos con técnicas de minería de datos, sino también los pasos previos, como el filtrado y preprocesado de los datos, y los posteriores, como la interpretación y validación del conocimiento extraído[18,19].

Normalmente el término minería de datos (Hernandez, Introducción minería de datos, 2001) lo usan estadísticos, analistas de datos, y la comunidad de sistemas de gestión de información, mientras que KDD es más utilizado en inteligencia artificial y aprendizaje en computadores.

el punto de vista de Fayyad señala los principales pasos dentro del proceso interactivo e iterativo del KDD :

1. Desarrollo y entendimiento del dominio de la aplicación, el conocimiento relevante y los objetivos del usuario final. En este paso se resalta la relación usuario/analista, a fin de identificar factores críticos

susceptibles de ser automático y cuáles no, cuáles son los objetivos, los criterios de rendimiento exigibles, para qué se usarán los resultados que se obtengan, compromisos entre simplicidad y precisión del conocimiento extraído, etc.

2. Creación del conjunto de datos objetivo, seleccionando el subconjunto de variables o ejemplos sobre los que se realizará el descubrimiento. esto implica consideraciones sobre la homogeneidad de los datos, su variación a lo largo del tiempo, estrategia de muestreo, grados de libertad, etc .
3. Preprocesado de los datos: eliminación de ruido, estrategias para manejar valores ausentes, normalización de los datos, etc.
4. Transformación y reducción de los datos. Incluye la búsqueda de características útiles de los datos según sea el objetivo final, la reducción del número de variables y la proyección de los datos sobre espacios de búsqueda en los que sea más fácil encontrar una solución. este es un paso crítico dentro del proceso global, que requiere un buen conocimiento del problema y una buena intuición, y que, con frecuencia, marca la diferencia entre el éxito o fracaso de la minería de datos (paso 7).
5. elección del tipo de sistema para minería de datos. esto depende de si el objetivo del proceso de KDD

es la clasificación, regresión, agrupamiento de conceptos (clustering), detección de desviaciones, etc. (en [Fayyad et al., 96] pueden verse en detalle los diferentes métodos de minería de datos)

6. elección del algoritmo de minería de datos.
7. Minería de datos. el proceso de minería de datos convierte datos en conocimiento. este paso será exitoso solo si los pasos previos se realizan correctamente[2,5].
8. Interpretación del conocimiento extraído, con posibilidad de iterar de nuevo desde el primer paso. La obtención de resultados aceptables dependerá de factores como: definición de medidas del interés del conocimiento (de tipo estadístico, en función de su sencillez, etc.) que permitan filtrarlo de forma automática, existencia de técnicas de visualización para facilitar la valoración de los resultados o búsqueda manual de conocimiento útil entre los resultados obtenidos.
9. Consolidación del conocimiento descubierto, incorporándolo al sistema, o simplemente documentándolo y enviándolo a la parte interesada. este paso incluye la revisión y resolución de posibles inconsistencias con otro conocimiento extraído previamente.

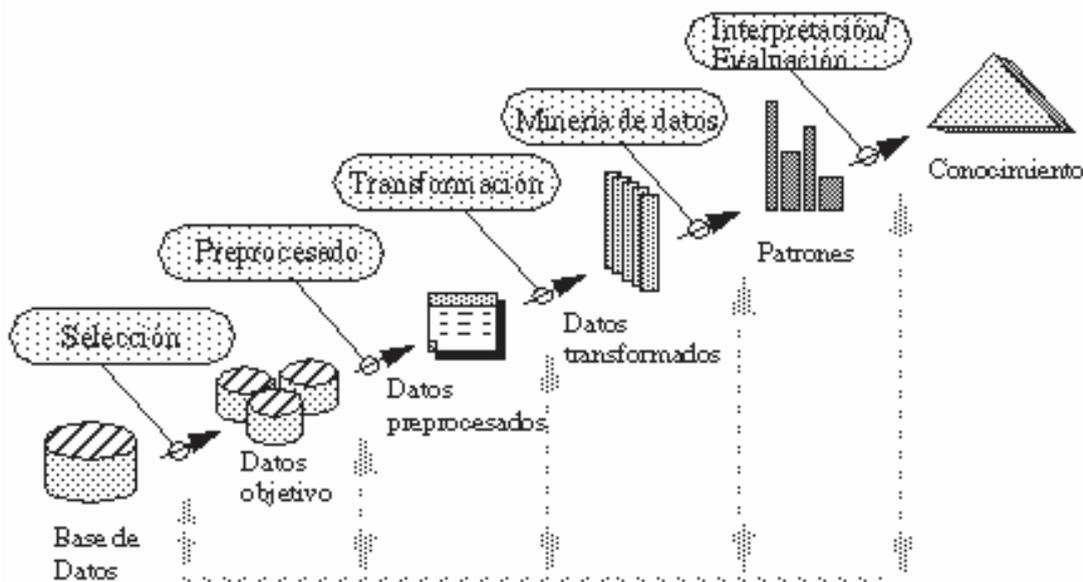


Figura 4. Fases para el proceso de generación de conocimiento.

2.3 Descubrimiento de Conocimiento Basado en Grafos

Puesto que en un sistema de descubrimiento de conocimiento basado en grafos, el algoritmo de minería de datos utiliza grafos como representación de conocimiento, la fase de preparación de datos incluye una transformación de los datos a un formato de grafo.

Espacio de Búsqueda:

el espacio de búsqueda de un algoritmo basado en grafos está compuesto de todos los grafos derivados a partir del grafo de entrada. De esto se deduce que el espacio de búsqueda es exponencial, de esto se desprende que el tiempo de ejecución de estos algoritmos es también exponencial.

Algunos Términos:

- Sea G un grafo. Definimos Subgrafo G' de G al grafo conectado cuyos vértices y arcos son subconjuntos de G .
- Subestructura S es un subgrafo que tiene asociada una descripción y un conjunto de instancias en el grafo de entrada.
- Una instancia es una ocurrencia de una subestructura s en un grafo G .

Representación de Conocimiento

el conocimiento desempeña un papel fundamental en las aplicaciones de la inteligencia artificial, y su representación sigue siendo un campo de continuo interés por parte de los investigadores en esta área. existen varias técnicas para representar el conocimiento de un dominio de la realidad en la computación inteligente.

ADARRAGA (Sistemas basados en conocimiento, colección, estructuras y procesos 1994, pag 154-156) señala que la estructura representacional del conocimiento más típico son las redes semánticas y las reglas de producción. Para ello utiliza los marcos como estructuras de información que representan clases de objetos en términos de propiedades y las relaciones existentes entre estos.

el método basado en grafos fue implementado en el sistema Subdue (Cook and Holder 1994). El modelo de representación que utiliza subdue es un grafo etiquetado. Los objetos se representan con vértices y las relaciones con arcos. Las etiquetas se utilizan para describir el significado de los arcos y vértices. Subdue es un sis-

tema de aprendizaje relacional utilizado para encontrar subestructuras (subgrafos) que aparecen repetidamente en la representación basada en grafos de bases de datos. Una vez que la base de datos está representada con grafos, subdue busca la subestructura que mejor comprime al grafo utilizando el principio MDL. Después de encontrar esta subestructura, subdue comprime el grafo y puede iterar repitiendo este proceso. subdue tiene la capacidad de realizar un mapeo inexacto que permite descubrir subestructuras con pequeñas variaciones [18,23].

Otra característica importante de subdue es que permite utilizar conocimiento previo representado como subestructuras predefinidas.

Cuando se trabaja con bases de datos relacionales, cada renglón se puede considerar como un evento y los atributos como objetos. Los eventos también se pueden ligar a otros eventos por medio de arcos. Los atributos de los eventos se describen mediante un conjunto de vértices y arcos, donde los arcos identifican los atributos específicos y los vértices especifican el valor de ese atributo para el evento. Una representación basada en grafos es lo suficientemente flexible para permitir tener más de una representación para un dominio dado, permitiendo al investigador experimentar para obtener la mejor representación para su dominio. La definición de los grafos tiene un formato específico que se da como entrada al sistema subdue. [3,4].

Algoritmo de búsqueda

Un sistema de búsqueda tiene dos componentes principales que son el espacio de búsqueda y la estrategia de control, o el algoritmo de búsqueda. el algoritmo inicia con un solo vértice como subestructura inicial y en cada iteración expande las instancias de aquella subestructura añadiendo un arco en cada posible manera. De esta forma genera nuevas subestructuras que podrían considerarse para expansión. el método de búsqueda también puede sesgarse utilizando conocimiento previo (p.e. subestructuras que creemos que pueden existir en los datos, pero que queremos estudiar con mayor detalle) dadas por el usuario (Cook and Holder 1994). En este caso, el usuario provee subestructuras de conocimiento previo como entrada a subdue. subdue encuentra instancias de las subestructuras de conocimiento previo en el grafo de entrada y continúa buscando extensiones de aquellas subestructuras [11,13]

```

Subdue (G: grafo, Limite, ContSubs, contador)
Inicio
    Ps = 0
    ListaPadres = Todas las estructuras de un
vértice
    Mientras ( PS ≤ Limite )
        Hijos = { }
        Mientras ( ListaPadres = {} )
            Padres =
            (Remove(ListaPadres)
            Instancia =
            Extender(Padres)
            ListaHijos =
            Grupo(Instancia)
            Evaluar (ListaHijos)
            Insertar subestructuras
            en listaHijos into Hijos mod
            Contsubs
            PS = PS + 1 //
            incrementa subestructuras
            Insertar Padres into
            MejorLista mod contador
            FinMientras
            ListaPadres = ListaHijos
        FinMientras
    return MejorLista
Fin

    Algoritmo de Búsqueda de subdue
    
```

Criterio de Evaluación

el algoritmo de minería de datos considerado utiliza criterio de evaluación para determinar cuales subgrafos del espacio de búsqueda son relevantes y pueden ser considerados como parte de los resultados. el método basado en grafos utiliza el principio de longitud de descripción mínima (MDL) para evaluar los subgrafos descubiertos.

sin embargo el principio MDL dice que la mejor descripción del conjunto de datos es aquella que minimiza la longitud de la descripción de todo el conjunto de datos. en el método basado en grafos, el principio MDL se utiliza para determinar que también un grafo comprime al grafo de entrada. De esta manera, todos los subgrafos que se generan durante el proceso de búsqueda

se evalúan de acuerdo al principio MDL y los mejores subgrafos se eligen como parte del resultado. el criterio de evaluación se utiliza para determinar cuales subgrafos del espacio de búsqueda son relevantes y pueden ser considerados como parte de los resultados. el principio MDL dice que la mejor descripción del conjunto de datos es aquella que minimiza la longitud de la descripción de todo el conjunto de datos.

4. CONCLUSIONES

se han expresado muchas opiniones en relación al aprendizaje y al conocimiento. es notable el consenso de opinión para un fenómeno de tan obvia importancia.

el criterio de evaluación que implementa subdue permite decidir cuales patrones se van a elegir como conocimiento importante o estructuras. Los argumentos que se han dado mas arriba, muestran que cada vez que encuentra una subestructura, subdue compara su valor con el de otras subestructuras, utilizando el principio Codificación Mínima (Minimum Encoding), una técnica derivada el Principio de Longitud de Descripción Mínima MDLP (Rissanen 1989), el cual dice que la mejor descripción de un conjunto de datos es aquella que minimiza la longitud de la descripción de todo el conjunto de datos. en relación a subdue, la mejor descripción del conjunto de datos es aquella que minimiza $I(S) + I(G|S)$, donde S es la subestructura utilizada para describir el grafo de entrada G, $I(S)$ es la longitud (en numero de bits) requerido para codificar S, e $I(G|S)$ es la longitud del grafo codificado G después de ser comprimido utilizando la subestructura s.

5. REFERENCIAS

- [1] Anderson James "Redes neuronales" Edit AlfaOmega México 2007
- [2] Del Brio Bonifacio, Redes neuronales y sistemas borrosos edit Alfaomega México 2007
- [3] Ian Sommerville, Ingenieria de software edit Pearson Addison Wesley Madrid 2005
- [4]. Hernández Orallo, Jose. Introducción a la minería de datos. edit Prentice Hall españa 2004.
- [5]. Pajares Martinsanz Gonzalo. Inteligencia artificial e ingeniería de conocimiento . edit AlfaOmega Mexico 2006
- [6]. Palma Méndez Jose. Inteligencia Artificial. McGRAW-HILL españa 2008

- [7] "Web site Usage Patterns Discovery Using a Graph Based Representation", Ivan Olmos and Jesus A. Gonzalez. IX Ibero-American Workshops on Machine Learning for Scientific Data Analysis, 2004. Pp. 345 – 354. November, 2004.
- [8] "subgraph Isomorphism Detection with a Representation based on Lists of Codes". Ivan Omos, Jesus A. Gonzalez, Mauricio Osorio. IX Ibero-American Workshop on Machine Learning for Scientific Data Analysis, 2004. Pp. 355 – 362. November, 2004.
- [9] "Using Graph-Based Concept Learning to the Predictive Toxicology Domain". Ivan Olmos and Jesus A. Gonzalez. Taller de Bioinformática y Biología Computacional, enc '04. Pp. 635 – 640, septiembre, 2004.
- [10] "subgraph Isomorphism Detection using a Code Based Representation". Ivan Olmos, Jesús A. Gonzalez and Mauricio Osorio. Proceedings of the 18th International FLAIRs Conference. Pp. 474 – 479. May, 2005.
- [11] "structural Relational Graph Based Data Mining Applied to the MultiFunctional spaces of Properties in Puebla of the Angels in the XVI XVII and XVIII Centuries". Oscar E. Romero, Jesus A. Gonzalez, Ivan Olmos, Rosalva Loreto. Poster session, FLAIRs 2005. Pp. 846 – 847.
- [12] "Identificación de secuencias de baja complejidad mediante el uso de algoritmos para representaciones graficas de genomas". González, Jesús A., Olmos, Ivan, Muños, Pablo, Vázquez, Candelario, Sánchez, Ma. Patricia. VI congreso nacional sobre biología molecular y celular de hongos, Octubre, 2005.
- [13] "Mining Common Patterns on Graphs". Ivan Olmos, Jesus A. Gonzalez, Mauricio Osorio. International Conference on Computational Intelligence and Security. Lecture Notes in Artificial Intelligence. Vol. 3802, 41-48, springer Verlag, 2005.
- [14] "Inexact Graph Matching: A case of study". Ivan Olmos, Jesus A. Gonzalez, Mauricio Osorio. To appear in Proceedings of the 19th International FLAIRs Conference. 2006.
- [15] "Defining new argumentation-based semantics by minimal models". Juan Carlos Nieves, Mauricio Osorio, Ivan Olmos, Ulises Cortes and Jesus A. Gonzalez. seventh Mexican International Conference on Computer Science (ENC'06). PP 210-220.
- [16] "Reduction between the subgraph Isomorphism Problem and Hamiltonian and SAT Problems". 17th Internacional Conference on electronics, Communications and Computers. Ivan Olmos, Mauricio Osorio, Jesús A. González. Febrero, 2007
- [17] "Descriptive Characteristics Generation and selection for Acute Leucemia Subtype Classification from Bone Marrow Digital Cells". M. Coral Galindo, Jesus A. González, Leopoldo Altamirano, Ivan Olmos. Research in Computer science, IssN 1870-4069, 2007.
- [18] "Minería de Datos Basada en Grafos". Ivan Olmos, Jesús A. Gonzalez y Mauricio Osorio. Visión Politécnica. Año 2, Num. 5, Enero - Junio, 2007.
- [19] "Métodos de Minería de Datos basada en Grafos". Ivan Olmos, Jesús A. González. 4º Encuentro de Investigación, INAOe. Pp. 277 – 280. 2003.
- [20] "Minería de Datos en Grafos Basada en una Representación de Listas de Códigos". Ivan Olmos, Jesús A. González. 5º Encuentro de Investigación, INAOe. Pp. 253 – 256. 2004
- [21] "Asociación Inexacta en Grafos". Iván Olmos, Jesús A. González, Mauricio Osorio. 6º Encuentro de Investigación, INAOe. Pp. 271-274. Noviembre, 2005.
- [22] Grafos: Herramienta informática para el aprendizaje y resolución de problemas reales de teoría de grafos Alejandro Rodríguez Villalobos X congreso de teoría de la organización Valencia 7 y 8 de setiembre de 2006
- [23] Aprendizaje Computacional eduardo Morales y Jesús González
<http://ccc.inaoep.mx/~emorales/Cursos/NvoAprend/grafos.pdf>