RNPS No. 2142
ISSN 2072-6287
Versión Digital

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

REPORTE TÉCNICO
Reconocimiento de Patrones

SERIE AZUL

Algoritmos de agrupamiento conceptuales: un estado del arte

Alejandro Guerra-Gandón, Sandro Vega-Pons
y José Ruiz-Shulcloper

RT_050 **junio 2012**

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

RNPS No. 2142
ISSN 2072-6287
Versión Digital

SERIE AZUL

REPORTE TÉCNICO
Reconocimiento de Patrones

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

Algoritmos de agrupamiento conceptuales: un estado del arte

Alejandro Guerra-Gandón, Sandro Vega-Pons
y José Ruiz-Shulcloper

RT_050 **junio 2012**

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

Algoritmos de agrupamiento conceptuales: un estado del arte

Alejandro Guerra-Gandón[1], Sandro Vega-Pons[1] y José Ruiz-Shulcloper[2]

[1] Dpto. Minería de Datos, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV),
La Habana, Cuba
{aguerra,svega}@cenatav.co.cu
[2] Dpto. Reconocimiento de Patrones, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV),
La Habana, Cuba
jshulcloper@cenatav.co.cu

RT_050, Serie Azul, CENATAV
Aprobado: 25 de mayo de 2012

Resumen. Los algoritmos de agrupamiento conceptuales surgen como una alternativa en la clasificación no supervisada o agrupamiento, para la solución de algunos problemas donde es necesario obtener los conceptos de los grupos resultantes. El presente trabajo muestra un panorama actualizado de los algoritmos existentes en la literatura y sus aplicaciones. De estos algoritmos se describen las fases de formación de la estructuración y formación de conceptos; además se realiza un análisis de las fortalezas y debilidades de cada uno de ellos. Es un punto de partida para cualquier especialista que desee adentrarse en el tema y en las conclusiones se propone una nueva línea de trabajo: la combinación de los resultados de algoritmos conceptuales.

Palabras clave: agrupamiento conceptual, clasificación no supervisada.

Abstract. Conceptual clustering emerges as an alternative unsupervised classification for the solution of some problems where it is necessary to get the concepts of the resulting groups. This work shows an updated overview of existing algorithms in the literature and their applications. It is a starting point for any specialist to start researching on the subject and in the conclusions we propose a new line of work: the combination of the results of conceptual clustering.

Keywords: conceptual clustering, unsupervised classification.

1 Introducción

La clasificación no supervisada o agrupamiento, es usualmente vista como el proceso de agrupar una colección de objetos, donde cada grupo contenga objetos que son más parecidos entre sí que a los objetos de otros grupos, de acuerdo con alguna medida de proximidad entre objetos. Su objetivo es dotar al usuario de un listado de los elementos que conforman cada uno de los grupos que se obtienen. En ocasiones el número de grupos es también un problema a resolver. En general, podemos enfrentarnos a dos tipos fundamentales de problemas de clasificación no supervisada: *libre*, en el caso que se desconozca el número de grupos a formar y *restringida* en el caso que este sea un dato del problema.

El enfoque de clasificación no supervisada plantea diferentes problemas. Dos problemas fundamentales son: (1) determinar las características de la proximidad que se debe utilizar para agrupar los objetos y(2) encontrar una buena medida de proximidad para resolver un problema en particular[1]. En los métodos clásicos de clasificación no supervisada la proximidad entre objetos se asume como una medida en un espacio multidimensional que opera sobre un conjunto fijo de atributos que caracterizan a los objetos. Los grupos se definen como una colección de objetos del espacio, con una alta proximidad intragrupo y una baja proximidad intergrupo. El enfoque clásico de la clasificación no supervisada tiene varias limitaciones.

La mayoría de las medidas de proximidad empleadas consideran a todos los atributos con igual importancia, por lo que no hacen distinción entre atributos que pueden ser más o menos relevantes para el problema a resolver[1]. Son escasos los mecanismos para seleccionar y evaluar atributos en el proceso de generación de los grupos, o sea, casi no se tienen en cuenta las consideraciones que emplean los métodos humanos para agrupar objetos. Observaciones de cómo las personas agrupan objetos indican que existe una tendencia a seleccionar uno o unos cuantos atributos relevantes y agrupar los objetos teniendo en cuenta solo estos atributos[1]. Cada grupo entonces contiene objetos que son similares teniendo en cuenta solo estos atributos “importantes”. Se espera que grupos diferentes tengan objetos con diferentes valores de estos atributos.

Los algoritmos clásicos tampoco permiten construir descripciones conceptuales de los grupos. Ellos dejan el problema de la interpretación de los grupos al analista. Esto es una gran limitación porque en muchas ocasiones el especialista está interesado no solo en determinar los grupos sino también en formular descripciones conceptuales de los mismos.

La idea de agrupar objetos en categorías descritas por conceptos fue introducida por Michalski en 1980. La técnica del agrupamiento conceptual está compuesta por dos tareas fundamentales: el agrupamiento de entidades en el que se determinan grupos a partir de una colección de objetos (estructuración), y la caracterización, en la cual se determina el concepto de cada grupo de la estructuración.

El problema que aborda los algoritmos conceptuales es, sin lugar a dudas, de una gran utilidad práctica. Consiste en no limitarse a darle al usuario el listado de los objetos que están en un mismo grupo y que por ende, se presume comparten características comunes, propiedades, al menos con un nivel superior al que lo hacen con los otros objetos que no están en ese grupo. Se pretende brindar más información, dar las propiedades que cumplen los grupos.

Resulta muy atractivo para un especialista de las áreas donde frecuentemente se hace necesario agrupar a los objetos para estudiar las propiedades que los caracterizan y distinguen, que al listado de los objetos se le añadiera una expresión, de preferencia en lenguaje natural, que expresara justamente una de las cosas que anda buscando: el significado de que ciertos objetos estén en el mismo grupo.

La clasificación no supervisada se apoya en una idea básica de la Teoría Clásica de Conjuntos: el concepto de conjunto es primario, no se define, sólo puede determinarse. Para ello hay dos procedimientos, uno denominado extensional, que consiste en dar la lista de los objetos que conforman al conjunto y otro denominado intencional, que consiste en dar la “propiedad” que caracteriza a los objetos de dicho conjunto. Los algoritmos clásicos de agrupamiento utilizan el primer procedimiento para dar los objetos que forman cada grupo mientas que los algoritmos de agrupamiento conceptual utilizan el segundo procedimiento.

En este reporte técnico se describen los principales algoritmos de agrupamiento conceptual. En la siguiente sección se presentan los conceptos principales de la clasificación no supervisada conceptual y se muestra una taxonomía construida a partir del estudio realizado. En las Secciones 3 y 4 se muestran los algoritmos conceptuales no incrementales e incrementales respectivamente. En la Sección 5 se mencionan algunas de las principales aplicaciones de los algoritmos conceptuales reportadas en la literatura. Finalmente en la Sección 6 se presentan las conclusiones del trabajo.

2 Teoría de algoritmos conceptuales

Sea $\mathbb{O} = \{O_1, O_2, \dots, O_n\}$ un conjunto de n objetos, $R = \{r_1, r_2, \dots, r_l\}$ un conjunto de l rasgos en términos de los cuales serán representados los objetos. Entenderemos por una representación del objeto i-ésimo O_i el l-uplo $X_i = (r_1(O_i), \dots, r_l(O_i)) = (x_{i1}, \dots, x_{il}) \in \Omega = \Omega_1 \times \dots \times \Omega_l$, donde $r_j(O_i) = x_{ij} \in \Omega_j$ es el valor que toma el rasgo r_j en el objeto O_i. El conjunto Ω_j es aquel que contiene los valores admisibles del rasgo r_j (dominio de definición de la variable). Denotaremos $X = \{X_1, X_2, \dots, X_n\}$ como una representación del conjunto de objetos $\mathbb{O}$, donde X_i es una representación de O_i.

Entre las representaciones de los objetos se define una función de semejanza Γ, la misma se define como una función que se aplica a un par de objetos O_i,O_j obteniendo como resultado un valor real que indica qué tan semejantes son dichos objetos. Formalmente se puede definir como $\Gamma: \mathbb{O} \times \mathbb{O} \to \mathbb{R}$. Existen dos tipos de funciones de semejanza: la similitud y la disimilitud. Si es una similitud cumple que mientras más grande sea el valor de $\Gamma(O_i, O_j)$, más semejantes son los objetos O_i y O_j. En cambio, si es una disimilitud mientras más pequeña sea la evaluación de $\Gamma(O_i, O_j)$, más semejantes son los objetos. Esta definición está dada de la forma más general posible, pues no existe un consenso sobre la misma. Existen diversos puntos de vista, los cuales a veces asumen el cumplimiento de determinadas propiedades como por ejemplo: la simetría, la no negatividad, etc., sin tener en cuenta realmente el tipo de los datos sobre los que se va a aplicar.

El producto cartesiano Ω es el espacio de representación inicial de los objetos. La naturaleza de las variables o atributos será cualquiera: univaluada (cualitativa nominal u ordinal, cuantitativa), no univaluada (conjunto o intervalo) o estructurada (tipo árbol o grafo, etc). Por lo tanto, sobre el espacio de representación inicial no supondremos ninguna estructura algebraica o topológica, es decir, sobre Ω_j no asumiremos ninguna operación algebraica o lógica, ni ninguna distancia (métrica) definida a priori. Estas operaciones y métricas podrían estar presentes pero no son necesarias.

El problema de la clasificación no supervisada consiste en hallar una estructura interna de un conjunto de descripciones de objetos en el espacio de representación, es decir, hallar una familia $P = \{C_1, \dots, C_m\}$, donde $C_i \subseteq \mathbb{O}$, $C_i \neq \emptyset$, tal que $\bigcup_{i=1}^{m} C_i = \mathbb{O}$. En ocasiones, será necesario que $C_i \cap C_j = \emptyset, i \neq j$ (partición). Así, el resultado obtenido depende, en una primera instancia, de la selección del propio espacio de representación inicial, además de una función de semejanza Γ y de un criterio de agrupamiento Π, el cual expresa la forma en que vamos a usar Γ.

Este criterio de agrupamiento es la razón por la cual un objeto va a pertenecer a un grupo o dos objetos o más pertenecerán al mismo grupo. Todo esto expresa que la selección de la función de semejanza y el criterio de agrupamiento son cruciales en la solución del problema de clasificación no supervisada y su determinación debe estar basada en el conocimiento del problema particular que se esté modelando.

El problema de la estructuración conceptual de espacios sobre $\mathbb{O}$ consiste en hallar los conceptos que describen a cada grupo correspondiente a una estructuración y representarlos de una forma natural e interpretable por los humanos. Para realizar esta tarea se define como función de semejanza Γ la cohesividad conceptual[2].

La cohesividad conceptual es una función que no depende solamente de las características de los objetos, sino también del lenguaje L usado para describir los grupos de objetos (conjunto de conceptos disponibles) y del medio E (conjunto de objetos vecinos), esto es:

$$\textit{Cohesividad Conceptual}\,(O_i, O_j) = f(O_i, O_j, L, E)\,. \tag{1}$$

Podemos decir que la cohesividad conceptual puede ser una similitud o una disimilitud en dependencia de su naturaleza, teniendo en cuenta las definiciones de estos dos conceptos expuestos en este trabajo. Su aporte principal es que para dos objetos, se tienen en cuenta no solo sus atributos, sino un poco más de información que puede contribuir a mejorar la calidad de la función. Por ejemplo, en la

Figura 2 se muestran dos puntos *A* y *B* que están "cercanos" entre sí. Sin embargo, una función de cohesividad conceptual debe tener en cuenta que pertenecen a configuraciones de puntos que representan conceptos diferentes. Un algoritmo de estructuración conceptual de espacios debería ser capaz de agrupar estos puntos en dos cuadrados, al igual que lo harían las personas.

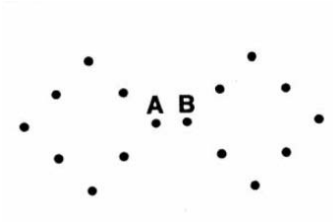

Fig. 1. Ejemplo de una distribución de puntos en $\mathbb{R}^2$. Un algoritmo conceptual debe ser capaz de agruparlos en las dos figuras geométricas.

Los algoritmos conceptuales o de agrupamiento conceptual se pueden dividir en dos grandes grupos: los algoritmos incrementales y los no incrementales, como se muestra en la Figura 3.

Los algoritmos incrementales basan su funcionamiento en la adaptación de los grupos (o conceptos) con los nuevos objetos que se le van presentando, es decir, cada vez que se analiza un nuevo objeto éste se clasifica, mediante cierta estrategia, en uno de los grupos ya existentes o se crean nuevos grupos. Por lo general, estos algoritmos se han desarrollado para los casos en que el conjunto de objetos a estructurar no está completamente dado, es decir, es dinámico. Ejemplos de algoritmos de este tipo son: UNIMEM [3], COBWEB [4] y Galois[5].

Por otro lado, los algoritmos no incrementales estructuran una muestra de objetos utilizando el conjunto de datos completo. Ejemplos de estos algoritmos son: CLUSTER/PAF [6], CLUSTER/2 [7], y CLUSTER/S [8], WITT [9], K-Means conceptual [10] y LCconceptual [11].

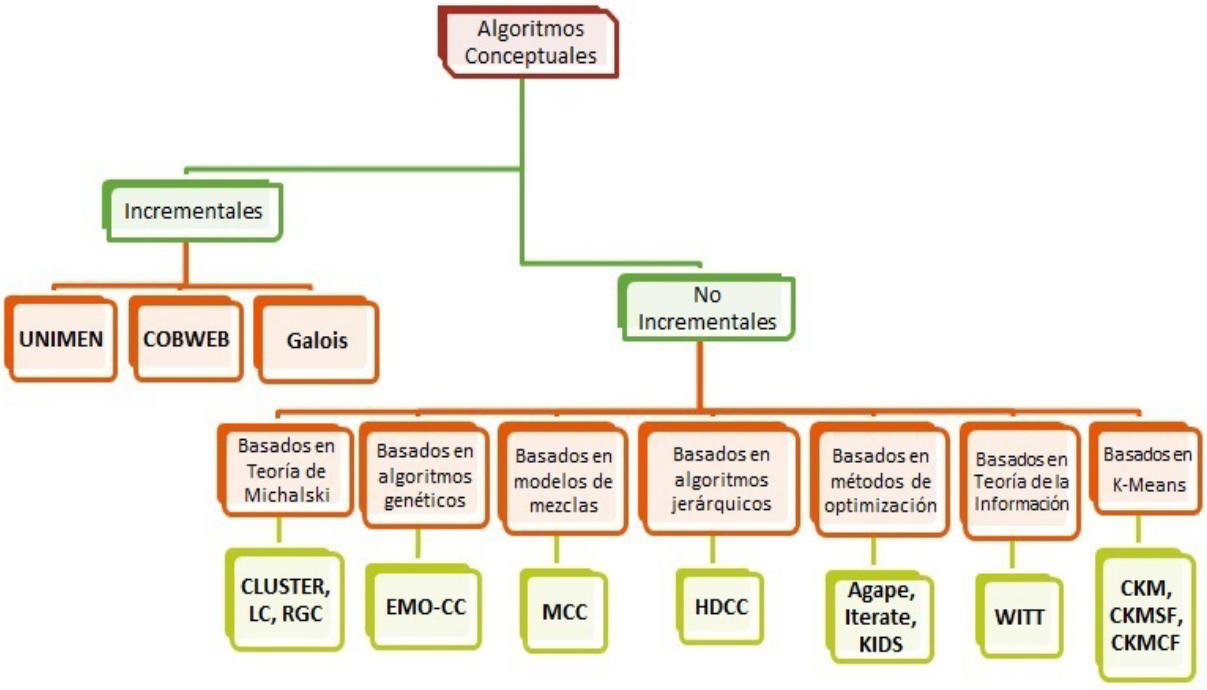

Fig. 2. Taxonomía de algoritmos conceptuales.

3 Algoritmos no incrementales

3.1 Algoritmos CLUSTER

En los trabajos de R. S. Michalski de finales de los años 70 y principios de los 80 se introdujo un conjunto de ideas que dieron origen al agrupamiento conceptual y a la familia de algoritmos CLUSTER, pioneros en la solución del agrupamiento conceptual. El primer miembro de esta familia fue CLUSTER/PAF, reconocido como el primer algoritmo conceptual. Este algoritmo fue posteriormente desarrollado, dando lugar a CLUSTER/2 y CLUSTER/S. La última versión del algoritmo es CLUSTER3[12], presentado en 2006.

Dado un conjunto de objetos no clasificados, *k* objetos semillas son seleccionados aleatoriamente y tratados como representativos individuales de *k* clases imaginarias. El algoritmo entonces genera descripciones de cada semilla que sean lo suficientemente generales y que no cubran ninguna otra semilla. Estas descripciones son usadas para determinar a los objetos más representativos de las nuevas clases formadas (los objetos que satisfacen la descripción de la clase). Los objetos representativos se usan como nuevas semillas en la próxima iteración. El proceso termina cuando iteraciones sucesivas convergen a la misma solución o cuando se rebasa un número específico de iteraciones sin mejorar la clasificación (según un cierto criterio).

Dados dos objetos O_i y O_j, la distancia sintáctica[2] entre dos objetos $\partial(O_i, O_j)$ está definida como el número de variables que tienen valores diferentes en O_i y O_j.

Una proposición relacional $[x_i \# R_i]$ donde $R_i \subseteq \Omega_i$, y $\#$ estandariza los operadores relacionales $\geq, >, \leq, <, =, \neq$ se denomina selector[2].

Un producto lógico de selectores es llamado un complejo lógico (l-complejo)[2]. Si los valores de las variables para el objeto O_i satisfacen todos los selectores en el l-complejo se dice que O_i satisface al l-complejo.

Cualquier conjunto de objetos para los cuales exista un l-complejo que satisface a estos objetos y sólo a estos objetos es llamado un conjunto complejo (*s*-complejo)[2].

Se dice que el l-complejo l_i está contenido en l_j y se denota $l_i \subseteq l_j$ si $s_i \subseteq s_j$ siendo s_i y s_j los s-complejos respectivos. Análogamente $l_i \cup l_j \Leftrightarrow s_i \cup s_j$ y $l_i \cap l_j \Leftrightarrow s_i \cap s_j$.

Sea $\Psi \subseteq \mathbb{O}$, un conjunto que se denomina de objetos observados, y los objetos en $\mathbb{O} \backslash \Psi$ (es decir, los objetos que están en el conjunto complemento de Ψ) son llamados objetos no observados. Sea α un l-complejo el cual cubre algunos objetos observados y algunos objetos no observados. El número de objetos observados en α es denotado por $p(\alpha)$. El número de objetos no observados cubiertos por α es llamado la *dispersión*[2] y es denotado por $s(\alpha)$. El número total de objetos en α es $t(\alpha) = p(\alpha) + s(\alpha)$.

La estrella $G(O_i|F)$ de un objeto O_i respecto a un conjunto de objetos F es el conjunto de todos los l-complejos maximales que cubren al objeto O_i y no cubren objeto alguno en F[2], ver Figura 4. (Un l-complejo α es maximal con respecto a una propiedad P, si no existe un l-complejo α'con la propiedad P, tal que $\alpha \subset \alpha'$.

Sean K_a y K_b dos conjuntos de objetos disjuntos, $K_a \cap K_b = \emptyset$. Un cubrimiento $COV(K_a|K_b)$ de K_a con respecto a K_b[2], es cualquier conjunto de l-complejos $\{\alpha_j\}_{j \in J}$, tal que para cada objeto $O_i \in K_a$ hay un l-complejo α_j, $j \in J$, cubriéndolo y ninguno de los l-complejos α_j cubre objeto alguno en K_b. Así se tiene lo siguiente: $K_a \subseteq \bigcup_{j \in J} \alpha_j \subseteq \mathbb{O} \backslash K_b$.

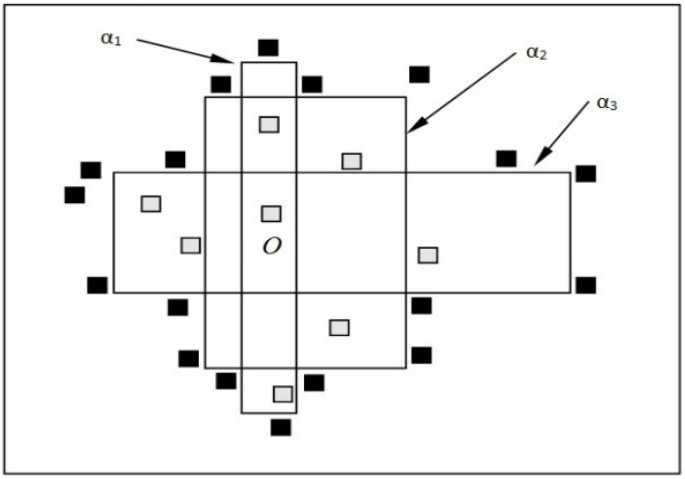

Fig. 3. Estrella de un objeto O con respecto al conjunto de cuadrados negros. Los l-complejos α_1, α_2 y α_3 son maximales con respecto a la propiedad de cubrir cuadrados grises.

La familia de algoritmos CLUSTER para la creación de los agrupamientos conceptuales, utiliza varios operadores:

- El operador REFUNION transforma un conjunto de objetos o l-complejos en un solo l-complejo tal que cubra a los objetos o l-complejos. Este operador determina para cada variable el conjunto de todos los valores que la variable toma en todos los objetos y l-complejos dados. Estos conjuntos son usados en la variable del nuevo l-complejo generado.
- El operador GEN simplifica y generaliza cualquier l-complejo dado aplicando una regla de generalización apropiada a cada selector en el l-complejo.

También utiliza los siguientes procedimientos:

- Procedimiento STAR: construye la estrella de un objeto O respecto a un conjunto de objetos F. Primero se construyen estrellas elementales para luego construir la estrella completa.
- Procedimiento REDUSTAR: los l-complejos en la estrella construida son reducidos y simplificados. La dispersión de cada l-complejo es reducida tanto como sea posible haciendo REFUNION de todos los objetos observados contenidos en cada l-complejo. Luego los l-complejos son generalizados y simplificados aplicando el operador GEN. El resultado es una estrella reducida.
- Procedimiento NID: transforma un conjunto de l-complejos no disjuntos en un conjunto de l-complejos disjuntos.

El criterio de calidad de un agrupamiento utilizado en la familia CLUSTER permite al usuario maximizar simultáneamente una o más medidas tales como:

- El ajuste entre los agrupamientos y los objetos: se calcula de dos formas diferentes, denotadas como *T* y *P*. La medida *T* es el negativo del total de la dispersión del agrupamiento, y la medida *P* es el negativo de la suma de la dispersión de los l-complejos.
- La simplicidad de las descripciones del agrupamiento: es definida como el negativo de la complejidad, el cual es el número total de selectores en todas las descripciones.
- La diferencia inter-agrupamientos: es la medida del grado de disyunción de cada par de l-complejos en el agrupamiento, el grado de disyunción de un par de l-complejos es el número de selectores en ambos l-complejos después de eliminar los selectores que se intersectan.
- El índice de discriminación: es el número de variables que solas discriminan entre todos los agrupamientos, esto es, variables que tienen valores diferentes en cada descripción del agrupamiento.
- La reducción de la dimensionalidad: es el negativo de la dimensionalidad esencial, definida como el mínimo número de variables requeridas para distinguir entre todos los complejos en un agrupamiento.

Estas definiciones son tales que el incremento del valor de cualquier criterio significa una mejora en la calidad del agrupamiento. La influencia relativa de cada criterio se especifica usando el Funcional de evaluación lexicográfica (LEF). LEF se define por una secuencia de pares criterio-tolerancia donde se incluyen criterios seleccionados de la lista anterior y para ellos se define un umbral de tolerancia [0…100%]. En el primer paso, todos los agrupamientos son evaluados con el primer criterio y aquellos que cumplen con el umbral son retenidos y evaluados por el siguiente criterio. El proceso continúa hasta que el conjunto de agrupamientos retenidos es reducido a uno (el mejor agrupamiento) o la secuencia de pares criterio-tolerancia es evaluada completa. En este último caso, los agrupamientos retenidos tienen calidad equivalente con respecto a LEF, y cualquiera puede ser seleccionado arbitrariamente. La selección de los criterios y su orden son definidos por un analista del problema.

En la Figura 5 se muestra el flujo de los algoritmos CLUSTER. En el paso donde se puede mejorar o no la calidad de la estructuración se propone utilizar para un caso semillas que sean objetos “centrales” y para el otro, semillas que sean objetos “bordes”. Los m objetos centrales son aquellos que, determinado por la distancia sintáctica, se encuentran más cerca del centro geométrico de los l-complejos. En cambio los objetos bordes, son aquellos que se encuentran más lejos.

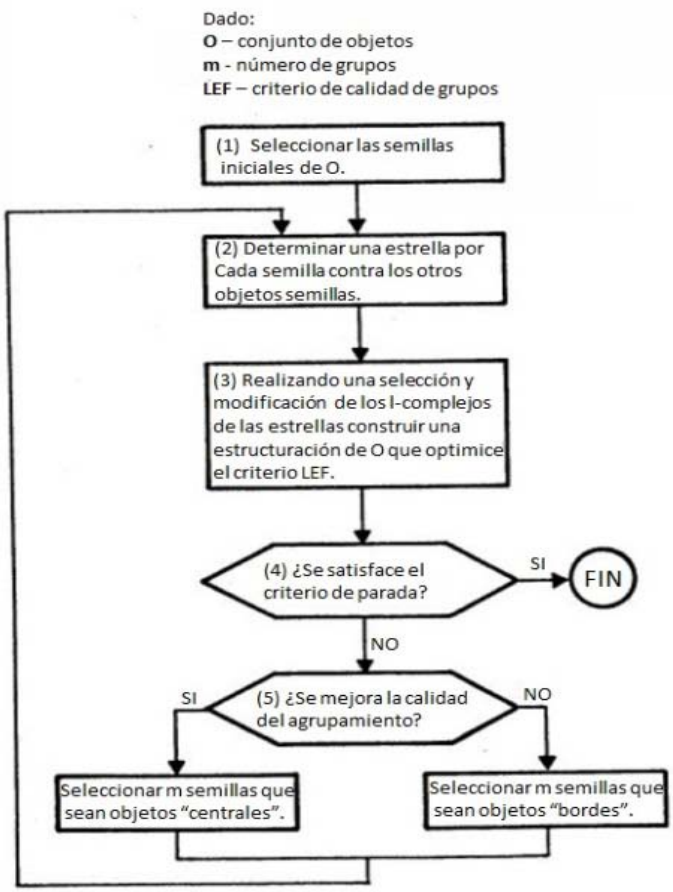

Fig. 4. Flujo de la familia de algoritmos CLUSTER.

3.1.1 CLUSTER3

Esta extensión de CLUSTER fue propuesta en el año 2006 por Michalski. La esencia del método es la misma, incorporando aspectos como la relevancia de atributos y modificaciones en la medida de calidad de los agrupamientos.

La relevancia de los atributos se tiene en cuenta de la siguiente forma. A diferencia de la clasificación no supervisada clásica, que plantea que su objetivo es encontrar la estructura que subyace en los datos, asumiendo que solo existe una, este algoritmo incorpora la idea de que pueden existir diferentes estructuras en el conjunto de datos en dependencia del punto de vista con que se quieran agrupar los mismos.

A continuación se expone un ejemplo donde se muestra un conjunto de atributos y su relevancia de de acuerdo a diferentes puntos de vista. En este caso el usuario debe definir los atributos que resultan relevantes para el problema a resolver y el complemento se califica como no relevante. Los atributos describen jugadores de hockey.

Tabla 1. Relevancia para diferentes puntos de vista de un conjunto de atributos para jugadores de hockey.

	Punto de vista		
Atributos	**Físico**	**Intelectual**	**Liderazgo**
Velocidad	Relevante		
Coeficiente inteligencia		Relevante	
Conocimiento de juego		Relevante	Relevante
Años de experiencia		Relevante	Relevante

Desde el punto de vista histórico, la familia CLUSTER tiene una gran importancia, pues con ella se introdujeron las ideas que dieron origen al agrupamiento conceptual. Estos algoritmos proporcionan los grupos junto con una descripción de las propiedades que lo caracterizan de una manera clara y de fácil comprensión para un ser humano.

El criterio de comparación para las variables es la igualdad, es decir dos valores son iguales si coinciden exactamente, este criterio es utilizado en la construcción de las estrellas y en la evaluación de la distancia sintáctica para la selección de las nuevas semillas en la nueva iteración del algoritmo. También debe resaltarse aquí que en muchos problemas prácticos el criterio de comparación de igualdad resulta muy estricto ya que existen situaciones en las que dos valores de una variable o dos descripciones de objetos pueden ser considerados como semejantes aunque no coincidan exactamente o totalmente.

Los algoritmos de la familia CLUSTER construyen propiedades o l-complejos de tal forma que sean disjuntos, lo cual en la práctica sin lugar a duda resulta de mucha utilidad, pero estos algoritmos no pueden resolver aquellos problemas prácticos donde se presenten situaciones en las que tenga sentido tener grupos no disjuntos y por tanto propiedades intersectadas, o en otras palabras, problemas donde los objetos pertenezcan a más de un grupo y por tanto cumplan más de una propiedad o concepto. La familia de algoritmos CLUSTER siempre crea particiones de la muestra original y deja sin solución aquellas situaciones en las que los grupos resultantes sean no disjuntos.

Se propone como método para evaluar la calidad de una propiedad (de un agrupamiento) el funcional de evaluación lexicográfico, este funcional considera distintos criterios para medir qué tan bueno es un agrupamiento, el resultado del mismo depende del orden en que estos criterios se apliquen y de los umbrales seleccionados para cada uno de ellos. En estos algoritmos no se proponen procedimientos para establecer tales valores ni el orden de aplicación de los criterios de calidad de un agrupamiento. Esta tarea la debe realizar el especialista del área de aplicación.

3.2 WITT

WITT fue propuesto por S. J. Hanson en 1989. Este algoritmo es un modelo computacional de la forma en que los humanos toman en cuenta las observaciones anteriores a la hora de agrupar objetos en categorías. La representación de los objetos y la adquisición del conocimiento se hacen empleando relaciones entre los rasgos. Este algoritmo de clasificación conceptual propone que la información para formar los grupos surge de la interrelación entre los atributos que describen a cada grupo y del contraste que cada uno de ellos tenga con los demás.

WITT presupone que los objetos están descritos en términos de rasgos cualitativos: booleanos y k-valentes. Este algoritmo centra su atención en las relaciones existentes entre los rasgos que caracterizan a los objetos y describe a los conceptos en términos de las coocurrencias entre pares de rasgos. Una de las herramientas que emplea WITT es la tabla de contingencia, pues ella ofrece una manera natural de representar tales relaciones.

Supongamos que r_i y r_j son atributos cualitativos que pueden tomar v_i y v_j valores diferentes (modalidades) respectivamente. Sea f_{mn} la frecuencia de coocurrencia del *m*-ésimo valor de r_i y el *n*-ésimo valor de r_j, es decir, el número de veces que se observa la combinación r_{i_m} con la r_{j_n}, m=1,…, v_i y n=1,…,v_j. Estas condiciones pueden representarse en una tabla, la cual recibe el nombre de tabla de contingencia.

Como metodología base para la construcción de los grupos, WITT utiliza una función de la teoría de la información para contrastar los grupos, tratando de maximizar la similitud dentro de cada grupo y minimizar la similitud entre grupos.

Sea un grupo C, se denomina *cohesión intragrupo* (de los objetos de C) y se denota por W_C, a la media de las varianzas de las co-ocurrencias de todos los posibles pares atributo-valor para dicho grupo y se calcula así:

$$W_C = \frac{\sum_{i=1}^{l-1}\sum_{j=i+1}^{l} D_{ij}}{\frac{l(l-1)}{2}}, \quad (2)$$

donde l es el número de rasgos y D_{ij} es la distribución de coocurrencias asociada a la tabla de contingencia de r_i respecto a r_j que viene dada por:

$$D_{ij} = \frac{\sum_{m=1}^{v_i}\sum_{n=1}^{v_j} f_{mn}\log(f_{mn})}{(\sum_{m=1}^{v_i}\sum_{n=1}^{v_j} f_{mn})log(\sum_{m=1}^{v_i}\sum_{n=1}^{v_j} f_{mn})}. \quad (3)$$

La función de cohesión intragrupo constituye una medida de similitud entre los objetos del grupo C, pues esta aumenta en la medida en que mayor sea la coocurrencia de los valores de dichos objetos.

Sean dos grupos C_p y C_t, se denomina cohesión relativa entre los grupos C_p y C_t a:

$$B_{C_p,C_t} = \frac{1}{W_{C_p} + W_{C_t} - 2W_{C_p \cup C_t}}. \quad (4)$$

Esta expresión mide la varianza de las coocurrencias en la unión de los dos grupos respecto a la de los dos por separado.

Sea un grupo C_p. Se denomina *cohesión del grupo* C_p a la expresión:

$$C_{C_p} = \frac{W_{C_p}}{O_{C_p}}. \quad (5)$$

donde W_{C_p} es la cohesión intra-grupo (de los objetos de C_p) y O_{C_p} representa la media de la cohesión del grupo C_p con el resto de los grupos existentes:

$$O_{C_p} = \frac{\sum_{t=1,t\neq p}^{m} B_{C_p,C_t}}{m-1}. \quad (6)$$

donde m es el número total de grupos y B_{C_p,C_t} es la cohesión relativa entre los grupos C_p y C_t.

La cohesión puede interpretarse como el contraste entre la media de la distancia de los objetos en el interior de un grupo respecto a la media de la distancia de ese grupo con el resto. Esta distancia tiene en cuenta la correlación entre los rasgos de los objetos, en contraste con la típica medida euclidiana, que asume la independencia entre ellos.

WITT parte de un conjunto de objetos definidos sobre un espacio de características (binarias o k-valentes). Los niveles aceptables de cohesión son indicados a través de tres parámetros que indican la coherencia relativa dentro y entre grupos.

Este algoritmo consta de dos fases: Inicialización o de pre-agrupamiento y Refinamiento. La primera fase opera formando un conjunto inicial de grupos o categorías tentativas mediante un algoritmo glotón ("greedy"). En ella no se utiliza la medida de cohesión para evaluar los grupos, sino una medida de distancia entre objetos o entre objetos y grupos. Este conjunto inicial se forma basado en las regiones densas del espacio de representación de los objetos, buscando los objetos más similares. Para ello, encuentra primero los dos objetos más cercanos y forma con ellos un grupo. Luego busca el próximo par de objetos más cercanos y crea un nuevo grupo, añade el objeto a un grupo existente o une dos grupos. Este algoritmo difiere de los algoritmos clásicos de taxonomía numérica en que emplea un

umbral (ξ_1) combinando los objetos hasta que la distancia entre los objetos más cercanos sea mayor que ξ_1. Al finalizar esta fase, se habrá formado un conjunto de grupos y podrán quedar objetos sin agrupar.

La segunda fase es más compleja, pues incorpora tres operadores: adicionar un objeto a un grupo existente, crear nuevos grupos y unir dos grupos. Es en esta fase de Refinamiento donde la medida de cohesión y las coocurrencias entre los rasgos desempeñan un papel fundamental.

La segunda fase comienza calculando la cohesión que resulta de añadir los objetos aún no clasificados en los grupos existentes. Luego, selecciona el par objeto-grupo de mayor cohesión y si ésta supera el umbral ξ_2 adiciona el objeto al grupo. Este proceso se repite hasta que la mayor cohesión no supere ese umbral. Esto sugiere que la estructuración obtenida es inadecuada, por lo que WITT intenta crear nuevos grupos invocando a la fase de pre-agrupamiento con los objetos no clasificados. Sin embargo, antes de adicionar los nuevos grupos a los existentes se asegura de que no ocupen el área en el espacio de los objetos de otro grupo ya existente. Esto se logra calculando $W_{C \cup C'}$ para cada grupo existente C y cada nuevo grupo C' y comparándolo con el umbral ξ_3. Si logró añadir nuevos grupos repite la fase de refinamiento para adicionar los objetos sin clasificar a los grupos existentes. Si todos los nuevos grupos se solapaban con los existentes, entonces WITT rechaza los nuevos grupos y considera, entonces, unir los grupos existentes, combinando el par de mayor cohesión intra-agrupamiento si ella supera el umbral ξ_3. Si esto ocurre, repite la fase de refinamiento para continuar incorporando los objetos que quedan sin clasificar.

Al terminar, WITT imprime los objetos pertenecientes a los grupos y las relaciones entre los rasgos (tablas de contingencia) de cada grupo.

En este algoritmo los grupos quedan caracterizados conceptualmente mediante el conjunto de las tablas de contingencia, una para cada par de rasgos, las cuales proporcionan una forma natural de representar las correlaciones entre los rasgos, es decir, en WITT los conceptos no son expresiones lógicas sino que están descritos en términos de las relaciones entre los rasgos.

La estructuración resultante es una partición del conjunto de datos inicial. El algoritmo no contempla la posibilidad de construir grupos solapados, a pesar de que el hombre en su proceso de categorización sí puede construir grupos de este tipo. Una dificultad de este algoritmo es que no garantiza agrupar a todos los objetos de la muestra inicial, pues al finalizar pueden quedar objetos que no estén ubicados en grupo alguno.

3.3 K-Means Conceptual

El algoritmo K-Means Conceptual, propuesto por Henri Ralambondrainy en 1995[10], es un método híbrido (numérico-simbólico) que integra una versión extendida del algoritmo K-Means[13] para la determinación de una partición de objetos descritos en términos de variables cuantitativas y cualitativas simultáneamente y un algoritmo de caracterización conceptual para la descripción intencional de los grupos.

El objeto $Juan = (masculino, Paris, profesor)$ se interpreta como el predicado:

$atributo(sexo, masculino) \wedge atributo(ciudad, Paris) \wedge atributo(trabajo, profesor)$.

Un retículo $L = (E, \leq, \wedge, \vee, *, \emptyset)$ es asociado con cada atributo $(r,\ D = \{d_1, \ldots, d_n\})$, donde E es el conjunto de subconjuntos de D que contiene todos los valores posibles que puede tomar el atributo r. El conjunto E es llamado el espacio de búsqueda asociado con el atributo r. El símbolo $\leq$ es la relación "es menos general que", la cual es un orden parcial definido como:

$$\forall e, f \in R, e \leq f \Rightarrow e \subseteq f\ . \tag{7}$$

Un retículo L tiene asociado un *máximo* elemento representado por $*$ y que es interpretado como "todos los posibles valores". También tiene un elemento *mínimo* $\emptyset$ (valor imposible). Cada par de atributos (e, f) tiene al menos un límite superior denotado por $e \vee f \in L$ y llamado "generalización" y un límite inferior denotado por $e \wedge f \in L$.

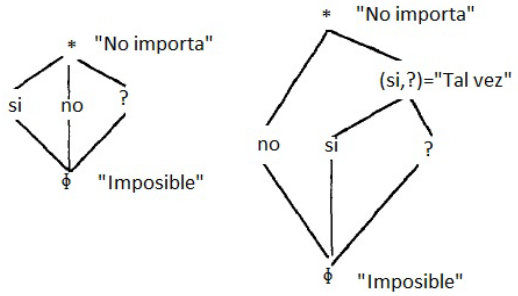

Fig. 5. Ejemplo de dos retículos para un mismo atributo.

El retículo es definido por el usuario a partir de la base de conocimiento disponible. Puede ser obtenido a través de cualquier estudio realizado sobre el conjunto de datos.

3.3.1 Paso de caracterización.

Para poder ser procesado un atributo numérico será codificado en uno simbólico. Denotemos por C el grupo que se desea caracterizar. Sea $\mathrm{media_C(r)}$ el promedio de los valores del atributo r en el grupo C y σ_r la desviación estándar de r. Un valor d de r es típico para C si:

$$media_C(r) - \sigma_r \leq d \leq media_C(r) + \sigma_r\,. \tag{8}$$

La función de codificación se define:

$$f: \mathbb{R} \to \mathcal{D} = \{inferior, típico, superior\} \tag{9}$$

$$f(d) = \begin{cases} \text{inferior}, & sid < media_C(r) - \sigma_r \\ \text{típico}, & simedia_C(r) - \sigma_r \leq d \leq media_C(r) + \sigma_r\,. \\ \text{superior}, & sid > media_C(r) + \sigma_r \end{cases} \tag{10}$$

El retículo asociado con una variable numérica tiene un espacio de búsqueda $E = \{inferior, típico, superior\}$.

Por tanto, a los atributos se les asocia un retículo y se representa por $\left(r_j, D_j, L_j\right)_{1\leq j\leq l}$.

El conjunto de objetos $\mathbb{O}$ también tiene asociada una estructura de retículo:

$$L = L_1 \times, \ldots, \times L_p = \varepsilon = \left(E_1 \times, \ldots, \times E_p, \leq, \vee, \wedge, *, \emptyset\right), \tag{11}$$

como el producto de los retículos L_j.

Tenemos entonces que $\forall \omega, \mathrm{o} \in \varepsilon: \omega = \left(\omega_\mathrm{j}\right)_{1\leq \mathrm{j}\leq \mathrm{l}}, \mathrm{o} = \left(\mathrm{o_j}\right)_{1\leq \mathrm{j}\leq \mathrm{l}}$ se cumple:

- $\omega \leq o \Leftrightarrow \left(\omega_j \leq o_j\right)_{1\leq j\leq l}$
- $\omega \vee o = \left(\omega_j \vee o_j\right)_{1\leq j\leq l}$
- $\omega \wedge o = \left(\omega_j \wedge o_j\right)_{1\leq j\leq l}$

Para $\mathrm{r_i(\omega)} \in \mathrm{E_i}: 1 \leq \mathrm{i} \leq \mathrm{l}$, se define el siguiente predicado: $\mathrm{A_{r_j(\omega)}}: \varepsilon \to \{\text{verdadero}, \text{falso}\}$ y dado por:

$\forall\omega' = \left(r_i(\omega')\right)_{1\le i\le l} \in \varepsilon: A_{r_i(\omega)}(w') = verdadero,\ sir_i(\omega') \le r_i(\omega)$ en otro caso se tiene que $A_{r_i(\omega)}(w') = falso$. Se asocia también a cada objeto $\omega = \left(r_i(\omega)\right)_{1\le i\le l}$ el siguiente predicado:

$$A_\omega = \wedge_{1\le i\le l} A_{r_i(\omega)}\,. \tag{12}$$

Un grupo C puede ser representado por el predicado $A_C = \vee\{A_\omega : \omega \in C\}$. La meta es aproximar A_C con un predicado A'_c que sea más general y más simple que A_C.

Sea ∂ la distancia definida por la diferencia simétrica entre funciones características. Se requiere entonces encontrar A'_c que minimice $\partial(A_c, A'_c)$ con $A'_c = \vee_{1\le j\le q} A_j$ bajo las condiciones:

- q mínimo (criterio de simplicidad)
- A_j más general que A_w (criterio de generalidad)

Llamemos ejemplos a los objetos que pertenecen al grupo C. El conjunto de contraejemplos es el complemento C' de C. Este criterio tiene la interpretación siguiente:

$\partial(A_c, A'_c) = |\text{contraejemplos}(A'_c)| + |\text{ejemplos}^*(A'_c)|$, siendo $|\text{contraejemplos}(A'_c)|$ el número de contraejemplos reconocidos por el predicado A'_c y $|\text{ejemplos}^*(A'_c)|$ el número de ejemplos no reconocidos por A'_c.

La selección de los predicados A_j presupone el cumplimiento de las condiciones:

$C1$: el predicado A_j debe ser α-discriminante, lo que quiere decir que la cantidad de contraejemplos reconocidos por A_j debe ser menor que α. $\left|contraejemplos\left(A_j\right)\right| \le \alpha$.

$C2$: el número de elementos del grupo C reconocidos por A_j debe ser mayor o igual que β. $\left|ejemplos\left(A_j\right)\right| \ge \beta$. A esto se le llama predicado β-caracterizador.

Se denota entonces por $EX^q_{\alpha-disc}$ el conjunto que contiene predicados α-discriminates generados por q objetos.

Por ejemplo:

$$EX^2_{\alpha-disc} = \{A_e \vee A_{e'} : e, e' \in O,\ \ e \ne e', A_e \vee A_{e'} es \alpha - discriminante\}\,. \tag{13}$$

En el paso de caracterización, para comenzar, un predicado inicial es construido para cada objeto. Basados en estos predicados y en el retículo de generalización son generados predicados generalizados. Dos predicados P_1 y P_2 son generalizados si para cada atributo los valores de los mismos son iguales o estos pueden ser generalizados en el retículo definido para el atributo.

La generalización de dos valores x_i y x_j de un atributo r se realiza de la siguiente manera:

- Si x_i y x_j son iguales el predicado toma este valor para r.
- Si los valores son diferentes y un valor es más general que el otro, entonces el valor de r en el predicado sería ese valor más general.
- Si los valores son diferentes entonces el valor de r en el predicado sería la generalización de x_i y x_j en el retículo.

Un predicado generalizado es almacenado si es α-discriminante y β-caracterizador, en otro caso el predicado es eliminado. El proceso de generalización se repite hasta que no se puedan generar más predicados o se cumpla un número de iteraciones.

3.3.2 Extensión CKMCF

Este algoritmo propuesto por Irene Olaya en el año 2006[14]. Es una modificación de K-Means Conceptual basado en atributos complejos [14]y también consta de dos fases, el paso de agregación y el paso de caracterización. Utiliza para la construcción de los conceptos los atributos complejos, los cuales son valores para un subconjunto de atributos tales que valores que aparecen en los objetos del grupo C_i no aparecen en los objetos de otros grupos.

Con el objetivo de obtener los atributos complejos, deben ser buscados subconjuntos de atributos, los cuales reciben el nombre de conjuntos soporte. Los siguientes conjuntos soporte son usados por CMKCF:

- Conjunto soporte μ-discriminante: la diferencia entre objetos de diferentes grupos es mayor que la diferencia considerando todos los atributos.
- Conjunto soporte μ-caracterizador: la similitud entre objetos del mismo grupo es mayor que la similitud considerando todos los atributos.
- Conjunto soporte μ-testor: son aquellos que satisfacen al mismo tiempo las dos propiedades anteriores.

Los conjuntos soporte se obtienen a partir de un algoritmo genético y a partir de ellos se construyen los atributos complejos.

Para generar los conceptos, un predicado se asocia a cada atributo complejo.

Entre las restricciones del algoritmo, tanto el original como su extensión, en el paso de caracterización, las variables numéricas son transformadas o codificadas para que tomen valores no numéricos. La función de codificación siempre es la misma para todas las variables numéricas, y no resulta claro por qué únicamente deben tomar tres valores, a saber *inferior*, *típico* y *superior*, si bien es cierto que para cada variable estos valores son calculados de manera distinta (es decir, cada variable tiene una media y una desviación estándar respecto a cada grupo), el hecho de trabajar con esos tres valores hace que se pierda la semántica de la variable en el problema específico a resolver.

Cada atributo nominal necesita un retículo de generalización definido por el usuario. En la práctica, no hay forma de realizar esta tarea de manera automática. Además de que no a todas las variables cualitativas se les puede asociar un retículo de generalización, un ejemplo de esto ocurre con la variable color, si esta variable toma como valores {rojo, verde, amarillo, azul} no puede decirse para muchas aplicaciones, por ejemplo, que rojo y verde son más generales que azul.

También necesita de la definición de los parámetros α y β para su funcionamiento.

3.4 ITERATE[15]

Algoritmo propuesto en el año 1998 por Gautam Biswas. Es una heurística y tiene tres pasos fundamentales:

- Obtener un árbol de clasificación usando la medida *category utlility*(CU), como criterio para agrupar los objetos.
- Extraer una *buena* partición inicial de los datos a partir del árbol de clasificación como punto de inicio para la búsqueda de los grupos finales.
- Iterativamente redistribuir los objetos a través de los grupos hasta encontrar grupos lo *suficientemente buenos*, cuando se cumpla cierta condición de parada.

La medida CU para una clase C_k se define como:

$$CU_k = P(C_k)\left\{\sum_i \sum_j \left[P\left(r_i = x_{ij} \middle| C_k\right)^2 - P(r_i = x_{ij})^2\right]\right\}. \quad (14)$$

Donde $P(r_i = x_{ij})$ es la probabilidad de que el rasgo r_i tome el valor x_{ij} y $P\left(r_i = x_{ij} \middle| C_k\right)$ es la probabilidad condicional de que $r_i = x_{ij}$ en la clase C_k. La utilidad de una partición llamada *partition score*(PS) sobre una partición de m grupos se define como el promedio de CU en la misma:

$$PS = \frac{\sum_{k=1}^{m} CU_k}{m}. \quad (15)$$

Para crear el árbol de clasificación del cual se debe obtener una partición inicial se utiliza un algoritmo de agrupamiento jerárquico divisivo. El algoritmo jerárquico utiliza PS para determinar si un objeto debe ser agrupado en uno de los grupos existentes o formar un nuevo grupo.

La partición inicial se extrae comparando el valor de CU de los nodos a lo largo del caminodel árbol de clasificación. Para cualquier camino de la raíz a las hojas este valor incrementa y luego disminuye.

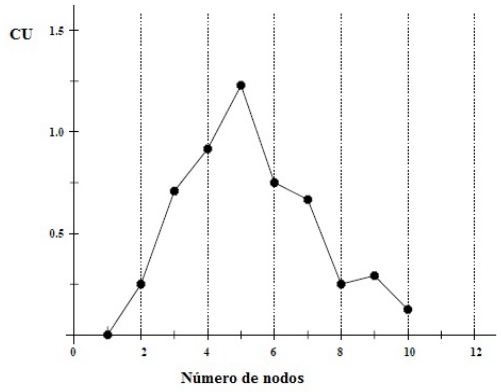

Fig. 6. Ejemplo de valores de CU a través del camino del árbol de clasificación.

Los nodos por debajo del nivel en que el valor de CU empieza a disminuir se considera que son información redundante. Esto implica que no sean utilizados para la tarea de agrupamiento.

La redistribución iterativa es aplicada luego con el propósito de maximizar la medida de cohesión para clases individuales en la partición. Esto consiste en asignar el objeto O_j a la clase k para la cual la medida *category match*(CM) es máxima. CMse define para un objeto O_j y un grupo C_k como sigue:

$$CM_{O_j C_k} = P(C_k) \sum_{i=1}^{l} \left[P\left(r_i = x_{ij} \middle| C_k\right)^2 - P(r_i = x_{ij})^2 \right]. \quad (16)$$

El proceso de redistribución se aplica hasta que no ocurran cambios.

El algoritmo es dependiente de la medida CM para la redistribución iterativa de los objetos en el proceso de clasificación, pero en lugar de aceptar los objetos de manera incremental, construye la estructuración inicial a partir de un algoritmo jerárquico. Se puede decir que es una adaptación no incremental del algoritmo incremental COBWEB.

No se argumenta por qué es redundante la información en los nodos a partir de que disminuye el valor de CU y tampoco se da la justificación por qué estos son los nodos que se deben utilizar en la construcción de la partición inicial.

Tiene también el problema del tamaño del árbol de clasificación, que en problemas reales su tamaño es muy grande y resulta complicado su interpretación.

3.5 LC

El algoritmo LC Conceptual fue propuesto en el año 2001 por Francisco Martínez[11] y está basado en los conceptos de la clasificación no supervisada en el enfoque lógico-combinatorio y retoma algunas ideas propuestas por Michalski para generar conceptos, en términos del conjunto de rasgos original.

Se dice que dos objetos$O_i, O_j \in \mathbb{O}$ son β_0-semejantes si $\Gamma(O_i, O_j) \geq \beta_0$, siendo Γ una función de semejanza. Si $\forall O_j \in \mathbb{O}$ se cumple que $\Gamma\left(O_i, O_j\right) < \beta_0$ entonces O_i se denomina β_0-aislado.

Un *criterio agrupacional duro* $\pi(\mathbb{O}, \Gamma, \beta_0)$ es un conjunto de proposiciones con parámetros O, Γ, β_0 tal que:

a) Genera una familia $P = \{C_1, \ldots, C_m\}$, $C_i \subseteq \mathbb{O}$ (grupos duros) que cumplen:

i. $\forall C_i \in P[C_i \neq \emptyset]$

ii. $\bigcup_{C_i \in P} C_i = \mathbb{O}$

iii. No existe C_r tal que $C_r \subseteq \bigcup_{t=1}^{k} C_{j_t}$, $C_{j_1}, \ldots, C_{j_k} \in P$

b) Define una relación $R_\pi \subseteq \mathbb{O} \times \mathbb{O} \times 2^{\mathbb{O}}$ ($2^{\mathbb{O}}$es elconjunto potencia de $\mathbb{O}$) que cumple:

iv. $\forall O_i, O_j \in \mathbb{O}\left[\exists C \in P, O_i, O_j \in P\right] \Leftrightarrow [\exists S \subseteq \mathbb{O}\ tal\ que\ R_\pi(O_i, O_j, S)]$

A cada C se le llama núcleo.

La familia P nos da el cubrimiento final. Un ejemplo de criterio agrupacional duro es el siguiente:

Sea $C \subseteq \mathbb{O}, C \neq \emptyset$. C es un núcleo β_0-conexo si se cumple que:

a) $\forall O_i, O_j \in C, O_i \neq O_j, \exists O_{i_1}, \ldots, O_{i_q} \in C[O_i = O_{i_1} \wedge O_j = O_{i_q} \wedge \forall p = 1, \ldots, q-1: \Gamma(O_p, O_{p+1}) \geq \beta_0]$.

b) $\forall O_i \in \mathbb{O}[O_j \in C \wedge \Gamma(O_i, O_j) \geq \beta_0] \Rightarrow O_i \in C$.

c) Todo elemento β_0-aislado es un núcleo β_0-conexo (degenerado).

Este criterio agrupacional es muy flexible, pues dos objetos pueden pertenecer al mismo núcleo y no parecerse. Otro criterio agrupacional que también genera particiones es el de núcleo $\beta_0 -$ compacto.

Sea $C \subseteq \mathbb{O}, C \neq \emptyset$. Ces un núcleo β_0-compacto si se cumple que:

a) $\forall O_j \in O\left[O_i \in O \wedge max_{\substack{O_s \in O \\ O_s \neq O_i}} \{\Gamma(O_i, O_s)\} = \Gamma(O_i, O_j) \geq \beta_0\right] \Rightarrow O_j \in C$.

b) $[max_{\substack{O_i \in O \\ O_i \neq O_p}} \{\Gamma(O_p, O_i)\} = \Gamma(O_p, O_t) \geq \beta_0 \wedge O_t \in C] \Rightarrow O_p \in C$.

c) $|C|$ es la mínima.

d) Todo elemento β_0-aisladoes un núcleo β_0-compacto (degenerado).

Otro concepto importante usado por el algoritmo LC es el concepto de testor. Para trabajar con los testores se requiere de una matriz de aprendizaje MA, es decir, una muestra inicial de objetos con la información de su pertenencia a las clases.

Un testor T es un subconjunto$\mathbb{R}' \subseteq \mathbb{R}$ tal que no existe alguna subfila (en términos de $\mathbb{R}'$) igual (β_0-semejante) a otra en clases diferentes. Al cardinal de T se le denomina longitud del testor.

Un testor típico es un testor en el que si se elimina un rasgo (columna) deja de ser testor.

El algoritmo LC consta de dos etapas: la de *estructuración extensional*, donde se forman los grupos y la de *estructuración intencional*, donde se caracteriza cada grupo mediante una propiedad lógica o concepto.

- En la primera etapa de estructuración extensional se utilizan los conceptos del enfoque lógico combinatorio para un problema de clasificación no supervisada. En ella se construyen los grupos de objetos basándose en la semejanza entre ellos y se utiliza un criterio de agrupamiento.
- En la segunda etapa de estructuración intencional o conceptual se construyen las propiedades (conceptos) que caracterizan a cada grupo de objetos.

El algoritmo LC-Conceptual se describe de la siguiente manera:

El número de testores típicos para ciertos problemas puede ser muy grande. Esto da como resultado que cada grupo podría tener asociado una gran cantidad de conceptos o propiedades, por lo cual, en el algoritmo se introduce un criterio para seleccionar al mejor o a los mejores conceptos. Un criterio podría ser el siguiente:

- Asociar a cada l-complejo un peso, el cual estará en función de los pesos informacionales de las variables calculados a partir de los testores típicos, es decir, a cada l-complejo α_t construido a partir del testor típico $t = \{x_{i1}, \dots, x_{is}\}$ se le asocia el peso $p(\alpha_t) = \sum_{j=1}^{s} p(x_{ij})$, donde $p(x_{ij})$ es el peso informacional de la variable x_{ij}.
- Como mejores l-complejos se toman aquellos que alcancen el máximo peso $p(\alpha_t)$. En general, podrían mostrarse al especialista ordenados de modo descendente por su peso y que él los seleccione.

En este algoritmo, los grupos se crean sobre la base de ciertas propiedades de la semejanza entre objetos y no sobre la base de criterios estadísticos o probabilísticos. En este algoritmo los grupos generados no son necesariamente particiones. Se obtendrán particiones o cubrimientos en dependencia del criterio agrupacional Π seleccionado.

Los conceptos construidos no son descripciones estadísticas de los grupos, que son difíciles de interpretar por los especialistas, sino propiedades lógicas basadas en los rasgos, en término de los cuales se describen a los objetos en estudio. Cada grupo tiene asociado uno o más conceptos que lo representan, tantos como testores típicos sean seleccionados.

3.6 RGC[16]

El algoritmo RGC lo propone Aurora Pons en el 2002 y responde a la siguiente idea: dado un conjunto de descripciones de objetos en términos de un conjunto de rasgos, el objetivo es encontrar una estructuración conceptual de estos objetos en el espacio de representación inicial.

Este algoritmo consta de dos etapas: una de determinación extensional y otra de determinación intencional. En la primera etapa, los grupos son creados usando alguna medida de similitud entre objetos y un criterio de agrupamiento para generar la estructuración. Para ello, podrá emplearse cualquier algoritmo de clasificación no supervisada. Luego, en la segunda etapa, los conceptos asociados a cada grupo son construidos y generalizados. En la Figura 11 se muestra el proceso del agrupamiento conceptual. Este algoritmo debe su nombre, precisamente, a las tres operaciones fundamentales realizadas en la determinación intencional: refunión extendida, generalización y construcción de los conceptos.

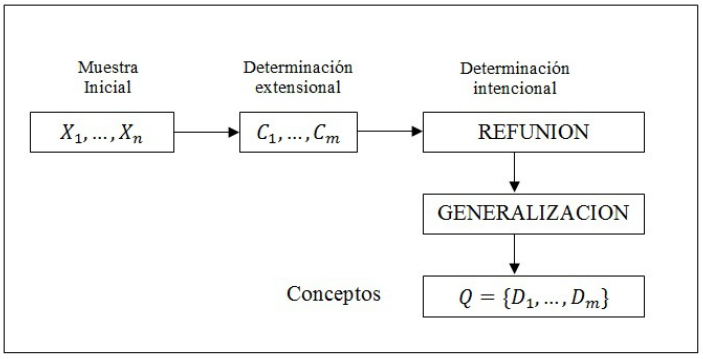

Fig. 8. Proceso del algoritmo RGC.

En la segunda etapa, se construyen los conceptos asociados a cada grupo obtenido en la primera. Primeramente, se selecciona un sistema de conjuntos de apoyo que deberá responder a las necesidades del problema. Para cada conjunto de apoyo y cada grupo se construirán los conceptos (l-complejos) que los caracterizan utilizando el operador de refunión extendida (*RUE*). Los l-complejos obtenidos mediante *RUE* cumplen la propiedad de ser excluyentes para cada grupo. Posteriormente, estos l-

complejos son simplificados y generalizados aplicando el operador *GEN* en cada variable de dichos l-complejos.

El operador *GEN* aumenta la dispersión de los l-complejos. A pesar de que al aplicar el operador *GEN* en cada variable del l-complejo se exige que no pierda su condición de ser excluyente para el grupo, al unir todas las generalizaciones de las variables en el l-complejo generalizado, puede perder esta propiedad.

Es por eso, que una vez generalizados los l-complejos es necesario determinar los objetos de los restantes grupos que los satisfacen y construir los conceptos.

El conjunto de l-complejos construido mediante el operador *RUE* pudiera ser diferente si se cambia el orden de presentación de los objetos. Esto no es importante, pues el objetivo es encontrar un concepto que caracteriza al grupo y no un único concepto. Además, resulta natural que un grupo no posea una única propiedad que lo caracteriza. La posibilidad de encontrar varias propiedades que caracterizan a un mismo grupo puede resultar, incluso, interesante para el especialista en el sentido de poder analizar los resultados desde distintos puntos de vista.

El algoritmo retoma las ideas de Michalki y realiza extensiones de varios operadores propuestos en los algoritmos CLUSTER para obtener descripciones de los grupos. Es un método para encontrar los conceptos de los grupos independientemente de la forma en que se construyeron los mismos, ya que propone utilizar cualquier algoritmo en la fase de agregación.

3.7 EMO-CC (Evolutionary Multiobjetive Conceptual Clustering)[17]

Algoritmo propuesto por R. Romero-Zaliz en el año 2006 y basado en la idea de los algoritmos evolutivos. Aplicado en el conjunto de datos de genes para recuperar subestructuras interesantes desde diferentes puntos de vista.

El método consta de cinco pasos fundamentales:

1. Representación del conjunto de datos. Se utiliza representación por grafos donde los nodos corresponden con los rasgos y las aristas con las relaciones entre ellos.
2. Estructura de aprendizaje. Para ello se utiliza NSGA II[18], un algoritmo evolutivo multi-objetivo. La configuración básica del mismo se explica a continuación:
   a) Representación de los cromosomas. EMO-CC codifica subestructuras factibles en los cromosomas. Cada cromosoma tiene forma de árbol donde cada nodo y arista se representa con una etiqueta que describe al tipo de atributo y un valor asociado al mismo. La población inicial es un conjunto de cromosomas, cada uno construido seleccionando un objeto aleatorio del conjunto de datos de entrada.
   b) Operadores genéticos. EMO-CC aplica los operadores genéticos cruzamiento y mutación con una probabilidad sobre los cromosomas que componen la población. El cruzamiento se realiza de la forma clásica, intercambiando dos subárboles de manera aleatoria. La mutación también se realiza de la manera clásica:
      i. Borrar una hoja: de manera aleatoria y solo con la arista que la une al árbol.
      ii. Cambiar un nodo: un nodo aleatorio es seleccionado y reemplazado por otro.
      iii. Adicionar una hoja: de manera aleatoria se crea una hoja y se conecta al árbol por una nueva arista.
   c) Selección. EMO-CC utiliza como selección el método clásico torneo binario[19], el cual escoge dos cromosomas padres y selecciona el de mayor valor de una función de calidad.
   d) Optimización multi-objetivo. Se considera que las buenas subestructuras son aquellas que maximizan un consenso de las funciones *specificity* y *sensitivity*. Por un lado, *specificity* está asociado al tamaño de la subestructura (el número de objetos y atributos que la forman). Mientras que *sensitivity* para una subestructura ses calcula a partir del número de instancias que ocurren en la subestructura. Una instancia ocurre en una subestructura si su representación por árbol es un subárbol del árbol que representa la subestructura.

e) Relaciones de dominio. Se seleccionan subestructuras que satisfagan el consenso entre su *specificity* y *sensitivity*, que son aquellas soluciones dominantes en el sentido que no hay otra solución superior a ella con respecto a las funciones objetivos. Otra función tenida en cuenta de forma implícita es la diversidad de la subestructura. Las relaciones de dominio con respecto a las funciones objetivos se tienen en cuenta de manera local, o sea, se trata de identificar subestructuras que sean las mejores en su vecindad. Se considera que dos subestructuras se encuentran en la misma vecindad si tienen al menos 50 % de las instancias en común basados en el coeficiente de Jaccard[20].

3. Evaluación. Se aplica la relación de dominio en una vecindad para obtener las subestructuras con las características deseadas.

4. Compresión de las subestructuras. Se realiza basándose en una consulta circunstancial, que permite obtener subestructuras flexibles y adaptables a diferentes contextos.

5. Inferencia. Se utiliza el clasificador no supervisado de los k prototipos más cercanos que caracteriza nuevas instancias basado en conocimiento disponible.

El algoritmo propone enfrentar el problema desde el punto de vista de los algoritmos genéticos. Es una extensión natural de este tipo de algoritmo para resolver el problema de los algoritmos conceptuales. Se incorpora la representación del conjunto de datos a través de grafos. La descripción mediante conceptos se realiza en función de una ontología de genes de acuerdo a la aplicación que muestran sus autores, no queda claro cómo se describirían los mismos en otra aplicación.

3.8 AGAPE[21]

Algoritmo propuesto por Julien Velcin en el año 2007. Su objetivo es extraer los conceptos optimizando una función de calidad global utilizando la metaheurística búsqueda tabú[22].

AGAPE es un método general que resuelve el problema de construir los grupos mediante la optimización de una función global. Esta función de calidad q debe ser definida de acuerdo al problema y debe computar la correspondencia entre el conjunto de objetos $\mathbb{O}$ y un grupo de conceptos dado $\mathcal{G}$. El objetivo es encontrar la solución $\mathcal{G}^*$ que optimiza q. Para ello se utiliza la metaheurística búsqueda tabú que mejora la búsqueda local básica.

La principal contribución de AGAPE es que descubre soluciones mejores de manera determinista escapando a óptimos locales. Además, se propone un nuevo operador basado en el algoritmo K-Means clásico. El método se implementó para la extracción de tópicos en conjuntos de datos de texto.

La función q que evalúa la calidad utiliza como entrada el conjunto de datos $\mathbb{O}$ y el conjunto de conceptos $\mathcal{G}$. Esta función brinda un valor en el intervalo $[0,1]$ y mientras mayor sea el mismo, mejor el conjunto $\mathcal{G}$ se corresponde con un conjunto de objetos dado.

Para el algoritmo los autores definen la siguiente función de calidad:

$$q(\mathbb{O},\mathcal{G}) = \frac{1}{|\mathbb{O}|}\sum_{O\in\mathbb{O}} sim(O, R_c(O))\,, \tag{17}$$

donde sim es una medida de similitud cualquiera y R_c es una función que relaciona el objeto O al concepto $d \in \mathcal{G}$ más cercano o que mejor lo describe.

En teoría el objetivo del algoritmo es encontrar el mejor conjunto de conceptos $\mathcal{G}^*$ tales que:

$$\mathcal{G}^*(\mathbb{O}) = \underset{\mathcal{G}\,\in\,\mathcal{H}}{argmax}\,\{q(\mathcal{G},\mathbb{O})\}\,, \tag{18}$$

donde $\mathcal{H}$ es el espacio hipotético, el conjunto de todos los posibles conjuntos de conceptos. Esta tarea se conoce que es un problema de elevada dificultad. Una buena aproximación $\mathcal{G}'$ de $\mathcal{G}^*$ se puede

encontrar a través de técnicas de optimización por lo cual se selecciona un algoritmo de búsqueda local, el cual utiliza la metaheurística de búsqueda tabú para salir de óptimos locales.

La búsqueda tabú explora el espacio hipotético $\mathcal{H}$ computando en cada paso una vecindad $\mathcal{V}$ de la solución actual. La vecindad contiene nuevas soluciones calculadas a partir de la solución actual mediante *movimientos* (operadores para cambiar de una solución a otra). La solución de $\mathcal{V}$ que optimice q se convierte en la nueva solución y el proceso se repite.

Para crear estas nuevas soluciones potenciales se consideran tres tipos de movimientos:

1. Dividir un concepto existente en p $(p > 1)$ nuevos conceptos.
2. Mezclar dos conceptos existentes para formar uno nuevo.
3. Ejecutar una variante de K-Means. Este movimiento está inspirado en el K-Means clásico y tiene dos pasos:
   a) Paso de ubicación. Asocia cada objeto O al concepto más cercano en $\mathcal{G}$ de acuerdo con la similitud definida y construye los grupos.
   b) Paso de actualización. Calcula una nueva descripción por cada concepto de acuerdo con los objetos cubiertos.

Este algoritmo necesita la definición previa de conceptos, con lo que en muchos problemas prácticos es difícil de contar, dado que los especialistas muchas veces no conocen los conceptos que describen a los objetos en el conjunto de datos. Muchas veces tampoco tiene sentido mezclar o dividir dos conceptos y no se aclara qué ocurre en caso de que esto suceda y se optimice la solución $\mathcal{V}$.

3.9 MCC (Model-based Conceptual Clustering)[23]

Algoritmo propuesto por Jeongkyu Lee en el año 2007 y basado en el algoritmo EM[24]. Es utilizado para la detección de objetos en video vigilancia.

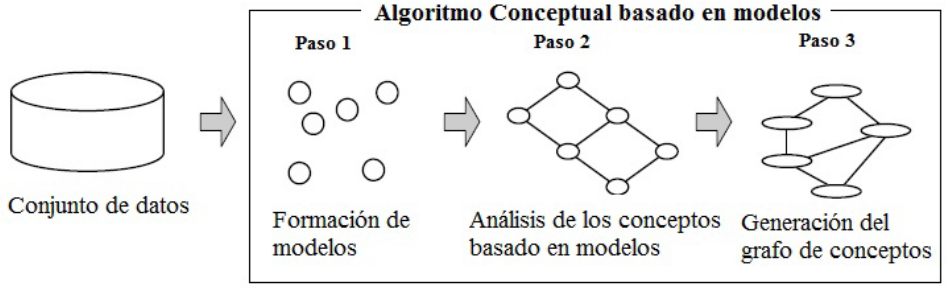

Fig. 9. Esquema de los tres pasos del algoritmo MCC.

El proceso de MCC consiste en tres pasos:

1. Formación del modelo. Se llama formación del modelo a la formación del agrupamiento donde cada grupo es llamado un modelo. Por tanto, a partir del conjunto de datos inicial se forma un conjunto de modelos utilizando el algoritmo EM. Cada modelo se caracteriza por un conjunto de parámetros usados para calcular el valor de relevancia de los atributos.
2. Análisis de los conceptos basado en modelos. Con el objetivo de extraer conceptos formales a partir de los modelos formados en el paso 1 se propone un análisis formal de conceptos basado en modelos (MFCA).

3. Generación del grafo de conceptos. En este paso se refinan los conceptos formales y las relaciones obtenidas en el paso anterior. MFCA generalmente produce un gran número de conceptos, por tanto es necesario agrupar conceptos similares en uno solo.

El resultado final de MCC es un grafo conceptual formado por un conjunto de nodos conceptuales y aristas relacionales.

Los beneficios de la formación de modelos utilizando EM son los siguientes:

a) Se simplifica el cómputo en el algoritmo pues el número de objetos es reducido al número de modelos que representan sus patrones específicos.

b) Se reutilizan algunos cálculos realizados por el algoritmo para establecer las relaciones entre los modelos y los rasgos.

El análisis de los conceptos basado en modelos se realiza calculando la relevancia del rasgo r_i sobre el modelo j mediante

$$\lambda(r_i, C_j) = \sum_{z=1}^{y} \left| P_{z,j,F} log \frac{P_{z,j,F}}{P_{z,j,F-\{r_i\}}} \right|, \tag{19}$$

donde y es la cantidad de objetos del grupo j. $P_{z,i,F}$ es definido por

$$P_{z,j,F} = P(y_{z,F} | C_{j,F}), \tag{20}$$

lo que representa la probabilidad de la tupla de rasgos del objeto z en el modelo j y $P_{z,j,F-\{r_i\}}$ es la probabilidad de la misma tupla en el modelo j sin tener en cuenta el rasgo i.

Un mayor valor de λ significa mayor relevancia del rasgo en el modelo. Todas las relaciones de relevancia pueden ser representadas en una tabla. Para eliminar relaciones que tienen valores poco significativos se predetermina un umbral.

El proceso de análisis de relevancia de los atributos es denominado MFCA. Como este proceso produce un gran número de conceptos, es necesario agrupar los conceptos similares en uno solo. Para esto se propone una medida de similitud entre conceptos llamada $ConSim$ expresada por

$$ConSim(D_1, D_2) = \gamma \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|} + (1 - \gamma) \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}. \tag{21}$$

$|A_1 \cap A_2|$ es la cantidad de objetos que tienen en común los grupos C_1 y C_2, mientras que $|B_1 \cap B_2|$ es la cantidad de conceptos en los que ambos grupos coinciden. De manera análoga para $|A_1 \cup A_2|$ y $|B_1 \cup B_2|$.

$ConSim$ toma valores entre 0 y 1. El mayor valor significa que los conceptos son más similares. El peso predefinido γ influye en la prioridad que se le quiere dar a los conceptos de los objetos o a la cantidad de objetos que forman los grupos.

Esta medida de similitud permite mezclar dos conceptos similares que luego son representados en un grafo conceptual. Dos conceptos son mezclados en un mismo nodo si el valor de $ConSim$ supera un valor predefinido.

Es un algoritmo muy novedoso, pues propone ideas como determinar la relevancia de rasgos para grupos y determinar la proximidad entre grupos en función de sus objetos y los conceptos que los describen. Propone además utilizar la información calculada en el paso de agregación, con el objetivo de usar toda la información valiosa que de que se disponga para el paso de caracterización. Tiene una aplicación muy útil en la video vigilancia.

3.10 HDCC (Hierarchical Distance-based Conceptual Clustering)[25]

Algoritmo propuesto por A. Funes en el 2008. Propone extraer los conceptos luego de construir el agrupamiento mediante algoritmos jerárquicos aglomerativos[26] (SingleLink, CompleteLink), entre otros. Es un método basado en distancias, ya que distribuye los elementos en los diferentes grupos basado en una medida numérica entre elementos.

La principal contribución de este método es que constituye una vía práctica y general para integrar métodos jerárquicos basados en distancias con algoritmos conceptuales.

Se realiza una adaptación que consiste en mezclar con cada nuevo grupo todos los grupos que son cubiertos por su generalización. De esta forma no se crean descripciones de grupos que cubran objetos de otros grupos. Además, los conceptos que describen los grupos finales dan una descripción de todos los elementos que están cercanos de acuerdo con la distancia utilizada pero también de aquellos que no están en el mismo grupo pero son cubiertos por el concepto.

Para representar el agrupamiento resultante se utiliza un dendrograma extendido llamado dendrograma conceptual. Este no solo brinda la información adicional referente a los objetos que están en cada grupo, sino también las descripciones de las propiedades comunes. En el mismo una línea continua enlaza los grupos unidos por la distancia utilizada y la línea discontinua la mezcla por conceptos.

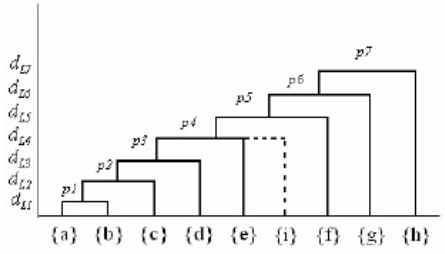

Fig. 7. Ejemplo del dendrograma extendido que utiliza HDCC.

Este algoritmo propone un acuerdo entre la formación de grupos teniendo en cuenta distancias y los conceptos que describen a los objetos. Para la representación de la unión de los grupos se utiliza el dendrograma extendido, una idea nueva incorporada en este algoritmo.

4 Algoritmos incrementales

4.1 UNIMEM

El algoritmo UNIMEM fue propuesto por Lebowitz[3], este algoritmo es de tipo incremental, es decir, acepta nuevos objetos uno a la vez para su clasificación.

UNIMEM considera que los objetos son descritos por una lista (pares atributo-valor) de variables de tipo cuantitativo y cualitativo. No se considera el manejo de ausencia de información en las descripciones de los objetos.

El algoritmo organiza conceptos en una estructura de memoria jerárquica denominada Memoria Basada en Generalización (*MBG*). La idea básica es que un proceso de generalización comienza a crear una jerarquía de conceptos que describe a un conjunto de objetos, y entonces el algoritmo graba en memoria información específica para tal fin.

Cada nodo en la estructura tiene como información: un conjunto de variables, un conjunto de objetos y apuntadores a nodos más específicos denominados subgeneralizaciones.

La descripción de un concepto se guarda en cada nodo de la jerarquía. Los nodos altos en la jerarquía representan conceptos generales, mientras que aquellos de bajo nivel en la estructura representan conceptos más específicos. La clasificación de los objetos está basada en las similitudes entre atributos.

Los conceptos son representados como conjunciones de pares atributo-valor, con cada valor de atributo teniendo un entero asociado que mide la predecibilidad, esto es, qué tan bien se puede predecir una característica dado un objeto del concepto.

Cada nodo o concepto puede tener un conjunto de objetos guardados en él, y un objeto puede ser guardado en múltiples nodos, así que la estructuración resultante no necesariamente es una partición de la muestra original.

El algoritmo UNIMEM involucra dos fases. La primera es una búsqueda recursiva en la estructura jerárquica existente *MBG*, para localizar el nodo o nodos (conceptos) más específicos con los que coincida el nuevo objeto a ser clasificado. Posteriormente el nuevo objeto se agrega a la jerarquía en cada uno de los nodos encontrados en la búsqueda previa. Esto incluye comparar el nuevo objeto con los ya guardados y si es conveniente generalizar creando nuevos nodos.

De manera general el algoritmo procede como sigue:

Dado un nodo se comparan los atributos de éste con los del nuevo objeto a clasificar. Para calcular el parecido entre el nuevo objeto y los nodos se utiliza la coincidencia total en el caso de atributos cualitativos y una función de distancia con umbral para atributos cuantitativos. Estos umbrales son parámetros del algoritmo. Además se usa un parámetro determinado por el usuario para poner el límite en el número de atributos en que el nuevo objeto a clasificar y el nodo deben ser parecidos. Con este método es posible que el objeto se clasifique en varias ramas diferentes de la jerarquía.

Tanto si coincide con los descendientes como si no, los valores de predecibilidad son modificados (aumentándolos o decrementándolos) teniendo en cuenta al nuevo objeto.

Si existen descendientes (subnodos) que coinciden con el nuevo objeto se sigue por el camino de los nodos que más se parezcan y que coincidan con los valores de los atributos del nuevo objeto.

Si ningún descendiente (subnodo) llega al límite de similitud se examinan los objetos almacenados bajo ese nodo.

Si alguno de estos objetos comparte suficientes valores con el nuevo objeto, dependiendo de otro parámetro de usuario, se crea un nuevo nodo generalizando los objetos parecidos y se almacenan estos objetos bajo el nuevo nodo. Cuando esto pasa, el algoritmo incrementa la predictibilidad de los atributos que aparecen en este nuevo nodo.

Si no hay ningún objeto suficientemente similar, se almacena el nuevo objeto bajo el nodo en curso.

La clasificación realizada por UNIMEM depende del orden en que los objetos son presentados, esto es claro dada la forma en que es construido el árbol debido a que los nodos se van creando en el proceso de generalización. Cuando en un cierto nodo se comparten suficientes valores (dependiendo de un parámetro del usuario) de las variables del nuevo objeto con alguno ya almacenado, se crea un nuevo nodo que generaliza a estos objetos. Esta estrategia creará nodos en dependencia del parecido de los objetos previamente clasificados y los nuevos objetos que van siendo clasificados. Se tiene entonces que el orden en que se presenten los objetos al algoritmo, determina la creación de los nodos (grupos) en la jerarquía dando como resultado diferentes estructuraciones.

La estructuración resultante puede ser una partición o un cubrimiento del conjunto de datos inicial, esto dependerá de la distribución de los objetos en los nodos durante el proceso de clasificación, ya que el cubrimiento se origina cuando un objeto nuevo se clasifica en dos o más nodos diferentes (debido a que el parecido con dichos nodos es igual).

Si la cantidad de objetos a agrupar y la cantidad de variables que los describen son grandes, lo que sucede con frecuencia en problemas prácticos reales, la *MBG* puede volverse inmanejable.

4.2 COBWEB

Este algoritmo fue propuesto por Douglas Fisher en el año 1987[4]. Una de las mayores motivaciones en la formación de conceptos es predecir características de los grupos. Si el grupo al que pertenece el objeto es conocido entonces conocemos sus propiedades porque los objetos heredan propiedades de los conceptos de más alto nivel. Esta conjetura forma la base de COBWEB.

Cuando los humanos observan el mundo, forman conceptos organizando sus observaciones, sobre la base de características compartidas entre las cosas que observan. Por ejemplo el concepto de perro es construido incrementalmente sobre el tiempo basándose en las observaciones de perros particulares de varias apariencias y comportamientos. El concepto crece en generalidad y empieza a ser más útil mientras uno encuentre más perros hasta que eventualmente se pueden hacer predicciones correctas acerca de un perro particular encontrado por primera vez. Esta categorización es realizada despreciando información irrelevante o incompleta.

De la misma forma que los humanos, COBWEB forma conceptos agrupando objetos con atributos similares. De esta forma se desprecia información irrelevante e incompleta en las nuevas observaciones u objetos que llegan. COBWEB es un algoritmo incremental, por lo que construye grupos examinando objetos uno a uno a la vez. Además representa grupos como una distribución de probabilidad sobre el espacio de valores de los atributos, generando un árbol de clasificación jerárquico. En el cual los nodos intermedios definen subconceptos. La versión original de COBWEB sólo permite trabajar con atributos cualitativos.

La tendencia de COBWEB es encontrar un conjunto de grupos que maximicen la utilidad de categoría (CU), ésta es una medida ideada para explicar el agrupamiento y el comportamiento de la formación de conceptos. Un grupo tiene alta utilidad si, dado el conocimiento de que el objeto O_i pertenece al grupo C_k, se puede predecir el valor x_{ij} del atributo r_j con alta exactitud. Una categoría además tiene utilidad alta si, dado el valor x_{ij} del atributo r_j para el objeto O_i (un miembro de C_k) es fácil determinar si O_i pertenece a C_k. Estas medidas son denominadas de predictibilidad y predictividad respectivamente.

La predictibilidad puede ser definida como la probabilidad condicional de que un objeto tenga un cierto valor de atributo dado el grupo (es decir, $P(r_j = x_{ij}|C_k)$). Similarmente la predictividad puede ser definida como la probabilidad condicional de que un objeto sea elemento de un grupo, dado el valor de un atributo particular (es decir, $P(C_k|r_j = x_{ij})$).

La CU se propone como un equilibrio entre la similitud intracategoría y la disimilitud intercategoría. La primera se refleja en la expresión $P(r_j = x_{ij}|C_k)$ donde $r_j = x_{ij}$ es un par atributo-valor y C_k es un grupo. Cuanto más grande es esta medida más predecible es el valor del atributo (dado un grupo). El segundo criterio es función de $P(C_k|r_j = x_{ij})$. Cuanto mayor es la probabilidad, más predictivo es el valor respecto a predecir la pertenencia al grupo.

Estas probabilidades se pueden combinar en un producto de forma que se establezca un balance entre ambas, generalizándolas para todos los valores de todos los atributos del dominio y para obtener una medida de calidad de una partición $\{C_1, \dots, C_q\}$.

La medida que resulta es

$$\sum_{k}^{n} \sum_{i} \sum_{j} P(C_k|r_j = x_{ij}) P(r_j = x_{ij}|C_k) . \tag{22}$$

Según Fisher, esta expresión establece un balance entre lo que denomina predecibilidad y predictividad. A esta medida se le añade una ponderación mediante la expresión $P(r_j = x_{ij})$ que establece que es más importante incrementar la predecibilidad y predictividad de los valores más frecuentes sobre los menos frecuentes, obteniéndose la expresión

$$\sum_{k}^{n}\sum_{i}\sum_{j} P(r_j = x_{ij})P(C_k|r_j = x_{ij})P\left(r_j = x_{ij}\middle|C_k\right). \quad (23)$$

A esta última expresión se le puede aplicar la regla de Bayes y obtener la siguiente expresión equivalente

$$\sum_{k}^{n} P(C_k)\sum_{i}\sum_{j} P^2\left(r_j = x_{ij}\middle|C_k\right). \quad (24)$$

Este nuevo término se interpreta entonces en función del potencial de inferencia de la partición. Así,

$$\sum_{i}\sum_{j} P^2\left(r_j = x_{ij}\middle|C_k\right). \quad (25)$$

es el valor esperado de valores de atributo que es posible conjeturar correctamente de un miembro arbitrario de C_k, suponiendo que un atributo se conjetura con una probabilidad igual a su probabilidad de ocurrencia y la probabilidad de que la conjetura sea correcta es la misma.

Finalmente, la utilidad de la estructuración se define como el incremento en el número esperado de valores, que pueden conjeturarse correctamente dada una partición, sobre el número de valores que pueden conjeturarse correctamente sin dicha información, dado por la expresión $P(r_j = x_{ij})$. Es decir, la CU para una partición $\{C_1, \dots, C_q\}$ es definida como:

$$CU = \frac{\sum_{k}^{n} P(C_k)\left[\sum_i\sum_j P^2\left(r_j=x_{ij}|C_k\right)-\sum_i\sum_j P^2(r_j=x_{ij})\right]}{n}. \quad (26)$$

El denominador es el número de grupos o clases de la partición, lo que permite comparar particiones de diferente tamaño.

La entrada a COBWEB es un conjunto de pares atributo-valor para cada objeto. COBWEB incrementalmente incorpora estas descripciones en un árbol de clasificación, donde cada nodo es un concepto probabilístico que representa un grupo de objetos. Los pares atributo-valor que se observan en los objetos que van llegando, son usados para calcular las probabilidades y la función CU.

Cada nodo en el árbol define un concepto probabilístico y provee un sumario de los objetos clasificados bajo este nodo. Un árbol de conceptos probabilístico difiere de un árbol de decisión en que es un descriptor probabilístico y no lógico lo que etiqueta cada nodo del árbol.

Para incorporar nuevos objetos en el árbol de clasificación, COBWEB desciende a lo largo del árbol un camino apropiado, actualizando el conteo de objetos a lo largo del camino, y realizando uno de varios operadores en cada nivel. Estos operadores incluyen: agregar un nuevo objeto a un grupo existente, crear un grupo nuevo, mezclar dos grupos en un solo grupo, y partir un grupo existente en varios grupos.

De manera general el algoritmo procede como sigue:

- Actualiza las probabilidades del nodo en curso según los valores del nuevo objeto.
- Si el nodo es terminal, el resultado es incorporar el nodo modificado, finalizando el algoritmo.
- Si el nodo no es terminal se evalúan las siguientes posibilidades según la función de CU, se escoge la mejor y se llama recursivamente a la función con el nodo en el que se haya decidido incorporar el nuevo objeto.

1. Se clasifica el objeto en cada descendiente del nodo en curso y se identifica el que maximice la función CU. Ese sería el nodo en el que se incorporaría el objeto (Incorporar).
2. Se calcula la función CU añadiendo un nuevo grupo que contenga únicamente al nuevo objeto.

3. Se une el mejor par de grupos (aquellos nodos que tengan la CU más grande al incorporar al nuevo objeto como se indica en 1.) y se incorpora el objeto a este nuevo grupo. Se escogería esta opción si se mejora la CU del nodo en curso (Mezclar).

Se particiona el mejor grupo y se añaden sus descendientes, calculando el resultado de la CU al añadir el nuevo objeto a cada uno de los grupos incorporados, dejándolo en el mejor.

Este algoritmo no utiliza una función de semejanza explícita, aunque implícitamente mide el parecido de un objeto con un grupo o nodo del árbol en función de la frecuencia de aparición de los valores de los rasgos del nuevo objeto en los objetos clasificados bajo el nodo, haciendo uso de probabilidades.

La construcción de la jerarquía de conceptos probabilísticos es dependiente del orden de presentación de los objetos, creándose, en general, diferentes jerarquías para órdenes diferentes de los datos.

Puede presentarse en algún momento el caso en que la utilidad de categoría alcance el mismo valor en más de una de las cuatro posibilidades de clasificación, esto significaría según la filosofía del algoritmo que hay dos posibles estructuraciones diferentes, en las cuales se puede clasificar al nuevo objeto. Sin embargo este algoritmo estructura al conjunto de objetos de la muestra en una partición, no se contempla la posibilidad de crear grupos traslapados. El caso en que la utilidad del grupo alcance el mismo valor en más de una de las cuatro posibilidades de clasificación se resuelve siguiendo el orden de las variantes de clasificación, así clasificar el nuevo objeto en una clase existente es lo primero que se verifica, después colocar al objeto en un nuevo grupo, mezclar dos grupos y finalmente partir un grupo existente.

De este algoritmo se ha realizado una extensión denominada DynamicWeb[27], el cual enfrenta el problema de la clasificación de objetos que cambian en el tiempo a través de múltiples observaciones. Su aplicación es en la seguridad de redes, donde un atacante a menudo cambia su información de identificación y el comportamiento con el fin de camuflar su comportamiento malicioso.

4.3 Galois

Algoritmo propuesto por Carpineto y Romano[5]. Permite determinar el retículo de conceptos correspondiente a un conjunto de objetos.

El mismo plantea dos ideas básicas:

- Para encontrar los conceptos en el nuevo retículo es suficiente considerar la intersección del nuevo objeto con los conceptos del retículo que se tiene hasta ese momento.
- El orden del retículo previo puede ser utilizado para la generación de cada nuevo concepto y sus aristas relativas mientras sea necesario. Esto evita el duplicado de conceptos y el chequeo de la consistencia de las aristas.

Para generar los conceptos en el nuevo retículo no es necesario interceptar el nuevo objeto con todas las posibles combinaciones de los objetos. Interceptando el nuevo objeto solo con los conceptos del retículo que se tiene formado hasta el momento es suficiente para garantizar que todos los nuevos conceptos que aporta este objeto sean generados.

Tabla 2. Descripción de los planetas del sistema solar.

	Tamaño			**Distancia del Sol**		**Luna**	
	Pequeño	**Medio**	**Grande**	**Cerca**	**Lejos**	**Si**	**No**
Mercurio	x			x			x
Venus	x			x			x
Tierra	x			x		X	
Marte	x			x		X	
Júpiter			x		x	X	
Saturno			x		x	X	

Urano		x			x	X	
Neptuno		x			x	X	

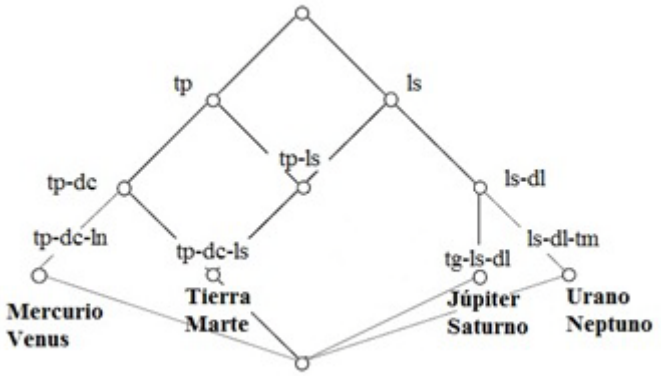

Fig. 8. Ejemplo de retículo para el conjunto de planetas descrito en la Tabla 1.

Una parte fundamental del algoritmo consiste en adicionar la intersección del nuevo objeto con los conceptos anteriores y sus aristas relativas al retículo.

Es muy costoso verificar que la intersección de un objeto con un concepto existente no haya sido generada anteriormente, si se quiere examinar cada nodo en el retículo. Por esto solo se compara los padres del nodo de la intersección con el nuevo objeto. Se pueden producir tres casos:

- El padre está incluido en la intersección. En este caso no se crea un nuevo concepto, es el mismo que se obtiene cuando se intercepta el nuevo objeto con el padre del nodo.
- El padre es igual a la intersección. En este caso no se crea un nuevo concepto porque ya está presente en el retículo.
- El padre no está incluido en la intersección o no se puede comparar con la intersección. En este caso se añade un nuevo concepto al retículo. Este procedimiento garantiza que nunca se generan dos conceptos iguales.

Luego de que un nuevo nodo se adiciona es necesario crear sus aristas relativas en el retículo. Para ello se sigue la siguiente estrategia: se determinan para cada nuevo nodo dos conjuntos limitantes, el conjunto límite inferior S y el conjunto límite superior G. Se procede entonces a enlazar el nuevo nodo con cada elemento en S y G y se eliminan las aristas entre S y G en caso de que existan.

El problema principal de los retículos es que su tamaño puede ser muy grande cuando se utilizan conjuntos de datos reales. Una forma de reducir el número de nodos es utilizar un lenguaje menos expresivo. AlphaGalois[28]es una extensión que propone una estrategia para disminuir el número de nodos.

Esta extensión del algoritmo Galois plantea agrupar el retículo por clases de equivalencia para su representación.

El algoritmo forma los grupos y los conceptos de los mismos en dependencia del orden en que se vayan presentando los objetos. Se basa en la idea de formar los conceptos basado en el retículo de atributos con el inconveniente de que este puede tener un gran tamaño en la resolución de problemas prácticos.

En el tema de los retículos se ha seguido investigando. Se ha trabajado en la búsqueda de patrones interesantes [29] y en retículos anidados [30]. Es importante resaltar que cualquier avance en este tema puede tener una aplicación directa en los algoritmos que utilizan el retículo para la representación de los conceptos. Otros estudios se han realizado en la visualización del retículo de conceptos [31] pues se requieren técnicas novedosas cuando es grande su tamaño.

5 Aplicaciones

Los algoritmos conceptuales tienen aplicaciones en muchos problemas reales. Entre las más recientes se encuentran la identificación de personas en redes sociales [32], el diagnóstico y la prevención de la diabetes[33], el reconocimiento gráfico de símbolos [34] y en la biología agrupando secuencias de ácido ribonucleico [35]. A continuación se explican con más detalle otras aplicaciones de estos algoritmos.

5.1 Agricultura[36]

Un problema importante es comparar cómo la clasificación obtenida a partir de un algoritmo conceptual se corresponde a la realizada por los especialistas humanos. Este problema ha sido tratado aplicando CLUSTER/2 y otras 18 técnicas a una versión del conjunto de datos *Soybean*. El objetivo del experimento fue ver cómo el algoritmo realizaba la clasificación de cuatro enfermedades. Los datos presentaban un total de 47 objetos descritos por 35 atributos.

Solo cuatro de los 18 métodos clasificaron correctamente sin equivocarse, sin embargo ninguno de ellos ofrece descripciones de los grupos formados.

CLUSTER/2 clasificó las enfermedades con un solo error y además construyó descripciones de las clases obtenidas. La figura muestra la descripción para una de las enfermedades (la cual es uno de los grupos formados) y se compara con la descripción de la misma brindada por un especialista del tema.

a) Descripción generada por CLUSTER/2	b)Descripción de un especialista
[Precipitation = Above normal] &	*[Precipitation = Normal or Above] &*
[Temperature = Normal] &	*[Temperature = Normal or Above] &*
[Stem Cankers = Above 2nd node] &	*[Stem Cankers = Above 2nd node] &*
[Canker Lesion Color = Brown or N/A] &	*[Canker Lesion Color = Brown] &*
[Fruiting Bodies = Present] &	*[Fruiting Bodies = Present] &*
[Condition of Fruit Pods = Normal] &	*[Condition of Fruit Pods = Normal] &*
[Time of Occurrence = Jul-Oct] &	*[Time of occurrence = Aug-Sep]*
[Damaged Area = Scattered or Low] &	
[Severity = Potential or Severe] &	
[Seed Treatment = None or Fungicide] &	
[Plant Height = Abnormal] &	
[Condition of Leaves = Abnormal] &	
[Leaf Spots = Absent] &	
[Shotholing/Shreading = Absent] &	
[Leaf Malformation = Absent] &	
[Leaf Mildew Growth = Absent] &	
[Condition of Seed = Normal] &	
[Condition of Stem = Abnormal] &	
[Extern. Stem Decay = Firm and Dry] &	
[Mycelium on Stem = Absent] &	
[Int. Discolor of Stem = None] &	
[Sclerotia int. or ext. = Absent] &	
[Condition of Roots = Normal] &	
[Plant stand = Normal]	

Fig. 9. Dos descripciones de un grupo de objetos en el conjunto de datos *Soybeans*[36].

Como se observa la descripción obtenida a partir de CLUSTER/2 se corresponde con lo descrito por el especialista, aunque además contiene otras condiciones adicionales que fueron observadas en los datos.

Esto es un resultado satisfactorio, pues muestra que CLUSTER/2 no solo clasifica correctamente, sino que también construye descripciones de los grupos acordes a las que realizan los especialistas.

5.2 Musicología[36]

A continuación se describe una aplicación de los algoritmos conceptuales utilizando CLUSTER/2 para determinar la clasificación de canciones españolas. El conjunto de datos construido por el musicólogo Pablo Poveda consiste en la descripción de 100 canciones españolas en función de 22 atributos.

La taxonomía generada se muestra en la figura. La misma divide las canciones en un total de 11 clases, cada una con entre 5 y 17 elementos.

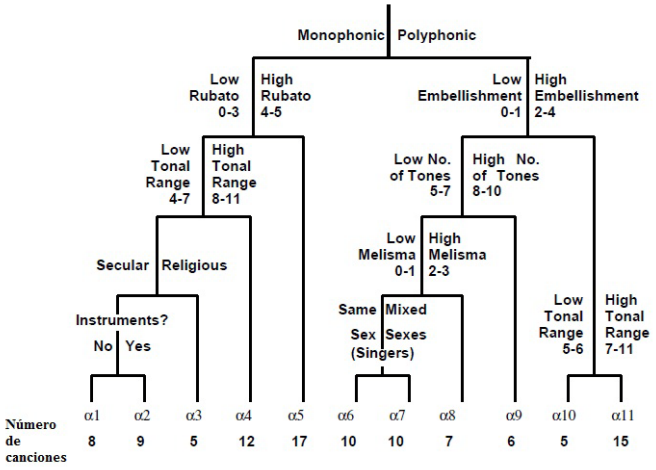

Fig. 10. Jerarquía de clasificación de canciones populares españolas[36].

Este resultado es de gran valor para los musicólogos, pues se corresponde bien con la forma en que se clasifican canciones, además brinda descripciones sencillas de las clases generadas por el algoritmo.

5.3 Detección de fraudes[36]

Esta aplicación combina la técnica de algoritmos conceptuales con la clasificación supervisada para descubrir reglas en la detección de fraudes en el cobro de impuestos.

En este enfoque, datos referentes a los impuestos son agrupados mediante un algoritmo conceptual y luego un método de clasificación supervisada es aplicado sobre los objetos en cada grupo con el objetivo de generar reglas simples que distingan objetos regulares y fraudulentos. En la figura se muestra el modelo utilizado en esta técnica.

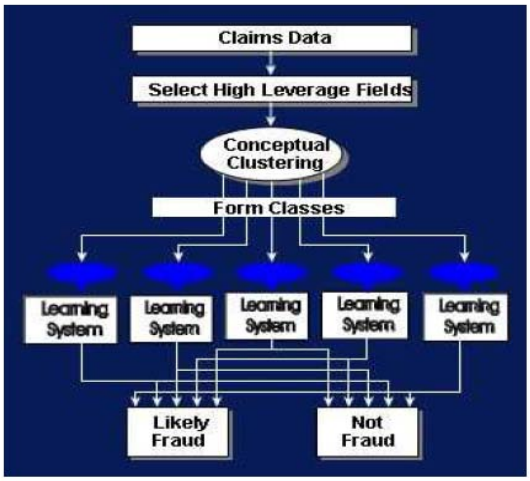

Fig. 11. Esquema del modelo para la detección de fraudes[36].

Luego estas reglas son aplicadas sobre los nuevos datos referentes a los impuestos. El experimento crea diferentes grupos de contribuyentes y entre ellos encuentra el grupo con un mayor porcentaje de violaciones con respecto a los otros.

5.4 Extracción de tópicos[21]

El algoritmo AGAPE se utiliza en el conjunto de datos reales AFP News, el cual fue extraído de la web francesa por el equipo de investigación de F. Chateauraynaud de manera automática. Contiene 1556 noticias descritas usando un vocabulario de 17 459 palabras o frases.

Se extrajeron 501 tópicos que pueden ser divididos en 14 principales, 186 débiles y 301 de valores atípicos. En la figura se muestran los primeros 7 tópicos principales.

t_1 (101 news): students, education, de Robien, academic, children, teachers, minister, parents, academy, high school ...
t_2 (93 news): AFP, Royal, UMP, Sarkozy, PS, president, campaigner, French, Paris, party, debate, presidential election ...
t_3 (73 news): project, merger, French deputy, bill, GDF, privatization, government, cluster, Suez ...
t_4 (67 news): nuclear, UN, sanction, USA, Iran, Russia, security council, China, Countries, uranium, case, enrichment ...
t_5 (49 news): rise, market, drop, Euros, analyst, index
t_6 (48 news): tennis, Flushing Meadows, US Open, chelem, year, tournament ...
t_7 (41 news): health minister, health insurance, health, statement, doctors, Bertrand, patients ...

Fig. 12. Los 7 tópicos principales del conjunto AFP[21].

5.5 Video vigilancia[23]

Los datos de videos reales fueron obtenidos de una cámara ubicada dentro de un local. Los principales movimientos que captaba eran de personas entrando por la puerta y caminando en la habitación.

Para generar los conceptos se aplicó MCC y en la figura se muestra el grafo conceptual resultante. Cada círculo y cada línea representan un nodo y una arista en el grafo respectivamente. El nodo

conceptual incluye un conjunto de modelos (descripción extensional) y un conjunto de atributos (descripción intencional). El número en la arista relacional indica la similitud entre dos nodos.

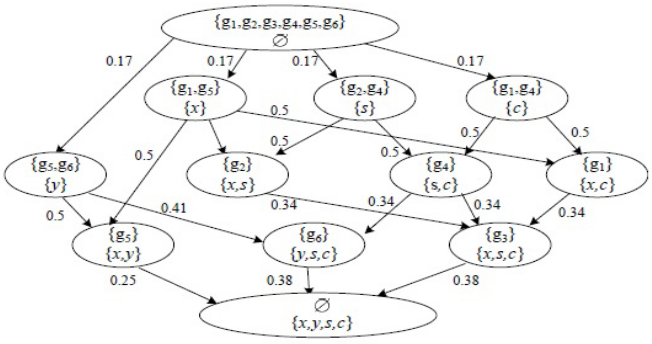

Fig. 13. Grafo generado a partir de un conjunto de datos reales[23].

6 Conclusiones

Los algoritmos conceptuales son uno de los paradigmas existentes en la actualidad dentro de la clasificación no supervisada. La gran cantidad de aplicaciones en diferentes campos de investigación, hace que el estudio de los métodos de agrupamiento conceptual sea de gran interés, como una nueva variante y distinta para enfrentar los problemas de clasificación.

El tema del agrupamiento conceptual como se pudo observar, tiene un gran impacto en la actualidad, debido principalmente a su capacidad de revelar estructuraciones de los datos y sus conceptos, lo cual libera al especialista de la interpretación de los resultados que en muchas ocasiones es engorrosa. A lo largo del trabajo se aprecian los métodos existentes desde los orígenes hasta la actualidad, mostrando algoritmos para conjuntos de datos dinámicos y estáticos, siendo estos últimos los que mayor desarrollo han alcanzado.

A lo largo del trabajo se aprecia que la gran mayoría de los algoritmos de agrupamiento conceptuales poseen parámetros en su propia definición. En algunos casos es necesario introducir el retículo de los atributos, la cantidad de objetos que debe describir un concepto, etc. por lo que se recomienda al igual que en el agrupamiento clásico, un estudio profundo del problema a resolver que posibilite mejores resultados.

Se ha comprobado que los algoritmos conceptuales tienen un mayor costo computacional que los clásicos y la razón fundamental es que cuentan con una etapa más, esta es precisamente donde se forman los conceptos de los grupos. Por su parte los algoritmos incrementales tienen la ventaja de que luego de que es construido el retículo para un conjunto grande de objetos, agregar un objeto nuevo tiene un bajo costo computacional, ya que no es necesario construir nuevamente el retículo de todos los objetos, pero tienen el inconveniente de la dependencia con el orden.

Un tema que no se ha estudiado con gran profundidad son los índices de validación para estructuraciones conceptuales, tanto internos como externos, por lo que se propone un futuro trabajo sobre el tema, donde se exponga lo existente en la literatura actual del tema. Este trabajo sería el paso inicial para el desarrollo de nuevos índices que permitan evaluar las estructuraciones conceptuales, que posibiliten realizar comparaciones entre los resultados de diferentes métodos y sirvan de base para un posible proceso de combinación de los resultados de este tipo de algoritmos.

Durante la realización de este trabajo se comprobó que no existen métodos para la combinación de los resultados de los algoritmos conceptuales. En los algoritmos clásicos y los algoritmos difusos[37] la combinación ha tenido muy buenos resultados: como promedio se mejoran los resultados de algoritmos

individuales; se elimina el ruido que pueda aportar el resultado de un algoritmo y se obtiene una estructuración que es un consenso de todas las implicadas en la combinación. No sería desacertado, por tanto, pensar que la combinación pudiera tener buenos resultados también para algoritmos conceptuales. Se podría utilizar para obtener estructuraciones duras combinando los conceptos o para obtener una estructuración conceptual de consenso y sería una nueva línea de trabajo inexplorada hasta el momento.

En resumen, se puede concluir que el agrupamiento conceptual emerge como una alternativa a la clasificación no supervisada que brinda distintos puntos de vista para el análisis de los datos. Pueden ser útiles en la resolución de determinados problemas prácticos y son una novedosa forma de descubrir la estructuración subyacente de un conjunto de objetos, por lo que es de gran interés el estudio y desarrollo de este tema en la actualidad.

Referencias bibliográficas

1. Michalski, R.S. and R.E. Stepp. Concept-based Clustering versus Numerical Taxonomy. in IEEE Transactions on Pattern Analysis and Machine Intelligence. 1981.
2. Michalski, R.S. and R.E. Stepp, Learning from Observation: Conceptual Clustering, in Machine Learning: An artificial intelligence approach 1983. p. 331-363.
3. Lebowitz, M., Experiments with Incremental Concept Formation: UNIMEM. Machine Learning, 1987. **2**(2): p. 103-138.
4. Fisher, D.H., Knowledge Acquisition Via Incremental Conceptual Clustering. Mach. Learn., 1987. **2**(2): p. 139-172.
5. Carpineto, C. and G. Romano, A lattice conceptual clustering system and its application to browsing retrieval. Mach. Learn., 1996. **24**(2): p. 95-122.
6. Michalski, R.S., Conceptual Clustering: A Theoretical Foundation and a Method for Partitioning Data into Conjunctive Concepts, 1979: In Textes des exposes du Seminaire organise par l'Institute de Recherche d'Informatique et d'Automatique (IRIA). p. pp. 254-294.
7. Michalski, R.S. and R.E. Stepp, Automated Construction of Classifications: Conceptual Clustering versus Numerical Taxonomy, 1983: IEEE Transactions on Pattern Analysis and Machine Intelligence.
8. Stepp, R.E. and R.S. Michalski, Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects, in Machine Learning: An Artificial Intelligence Approach, J.G.C. In R. S. Michalski, and T. M. Mitchell, Editor 1986, Morgan Kaufmann. p. pp. 471-498.
9. Jose, S., Hanson, and M. Bauer, Conceptual Clustering, Categorization, and Polymorphy. Mach. Learn., 1989. **3**(4): p. 343-372.
10. Ralambondrainy, H., A conceptual version of the K-means algorithm. Pattern Recogn. Lett., 1995. **16**(11): p. 1147-1157.
11. Martínez-Trinidad, J.F. and G. Sánchez-Díaz, LC: A Conceptual Clustering Algorithm, in Proceedings of the Second International Workshop on Machine Learning and Data Mining in Pattern Recognition 2001, Springer-Verlag. p. 117-127.
12. Seeman, W.D. and R.S. Michalski, The CLUSTER/3 system for goal-oriented conceptual clustering: method and preliminary results, 2006: Proceedings of The Data Mining and Information Engineering 2006 Conference, Prague, Czech Republic. p. 81-90.
13. MacQueen, J. Some methods for classification and analysis of multivariate observations. in Fifth Berkeley Symposium on Math. Stat. and Prob. 1967. University of California Press.
14. Ayaquica Martínez, I.O., J.F. Martínez Trinidad, and J.A. Carrasco Ochoa, Restricted Conceptual Clustering Algorithms based on Seeds. Computación y Sistemas, 2007. **11**: p. 174-187.
15. Gautam, B., et al., Iterate: A conceptual clustering algorithm for data mining, 1998: IEEE Transactions on Systems, Man and Cybernetics. p. 100-111.
16. Pons-Porrata, A., J. Ruiz-Shulcloper, and J.F. Martínez-Trinidad, RGC: a new conceptual clustering algorithm for mixed incomplete data sets, 2002: In Mathematical and Computer Modelling. p. 1375-1385.
17. Romero-Záliz R., et al. Mining Structural Databases: An Evolutionary Multi-Objective Conceptual Clustering Methodology. in Proceedings of the 4th European Workshop on Evolutionary Computation and Machine Learning in Bioinformatics. 2006. Budapest, Hungary.
18. Deb, K., et al., A fast and elitist multiobjective genetic algorithm : NSGA-II. Evolutionary Computation, IEEE Transactions on, 2002. **6**(2): p. 182-197.

19. Blickle, T. and L. Thiele, A Mathematical Analysis of Tournament Selection, in Proceedings of the 6th International Conference on Genetic Algorithms 1995, Morgan Kaufmann Publishers Inc. p. 9-16.
20. Hur, A., A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. in Pacific Symposium on Biocomputing. 2002.
21. Velcin, J. and J.-G. Ganascia, Topic Extraction with AGAPE, in Proceedings of the 3rd international conference on Advanced Data Mining and Applications 2007, Springer-Verlag: Harbin, China. p. 377-388.
22. Fred, G., L. Manuel, and M. Rafael, Tabu Search, 1997, Kluwer Academic Publishers.
23. Lee, J., P. Rajauria, and K.S. Subodh. A model-based conceptual clustering of moving objects in video surveillance. in Proceedings of SPIE, the International Society for Optical Engineering A. 2007. San Jose CA, USA.
24. Dempster, A.P., N.M. Laird, and D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1977. **39**(1): p. 1-38.
25. Funes, A.M., et al., Hierarchical Distance-Based Conceptual Clustering, in Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I 2008, Springer-Verlag: Antwerp, Belgium. p. 349-364.
26. Day, W.H.E. and H. Edelsbrunner, Efficient Algorithms for Agglomerative Hierarchical Clustering Methods. Journal of Classification, 1984. **1**: p. 7-24.
27. Scanlan, J.D., DynamicWEB: A Conceptual Clustering Algorithm for a Changing World, in Faculty of Science, Engineering and Technology (2011), University of Tasmania.
28. Ventos, V., H. Soldano, and T. Lamadon, Alpha Galois Lattices, in Proceedings of the Fourth IEEE International Conference on Data Mining 2004, IEEE Computer Society. p. 555-558.
29. Encheva, S., Lattices and patterns, in Proceedings of the 10th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases 2011, World Scientific and Engineering Academy and Society (WSEAS): Cambridge, UK. p. 156-161.
30. Encheva, S., Interval data and nested lattices, in Proceedings of the 10th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases 2011, World Scientific and Engineering Academy and Society (WSEAS): Cambridge, UK. p. 172-175.
31. Soto, M., B. Grand, and M.-A. Aufaure, Spatial Visualization of Conceptual Data, in Classification and Multivariate Analysis for Complex Data Structures, B. Fichet, et al., Editors. 2011, Springer Berlin Heidelberg. p. 379-388.
32. Indi, T.S., R.B. Kulkarni, and S.K. Dixit, Empirical Investigation of Finding Person in Social Network using Clustering. International Journal of Computer Applications, 2011. **19**(5): p. 29-34.
33. Suh, S.C. and G.P. Vudumula. The role of conceptual hierarchies in the diagnosis and prevention of diabetes. in 7th International Conference on Networked Computing and Advanced Information Management. 2011. Gyeongju
34. Boumaiza, A. and S. Tabbone. A Novel Approach for Graphics Recognition Based on Galois Lattice and Bag of Words Representation. in International Conference on Document Analysis and Recognition. 2011. Beijing
35. Baridam, B.B. and O. Owolabi, Conceptual Clustering Of RNA Sequences With Codon Usage Model. Global Journal of Computer Science and Technology, 2010. **10**(8): p. 41-45.
36. Ryszard, S.M., et al., Natural Induction and Conceptual Clustering: A Review of Applications, 2006: Reports of the Machine Learning and Inference Laboratory, George Mason University, Fairfax, VA.
37. Oliveira, J.V.d. and W. Pedrycz, Advances in Fuzzy Clustering and its Applications 2007: John Wiley & Sons, Inc.

RT_050, junio 2012
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2142
ISSN 2072-6287
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. No. 21812 e/218 y 222, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba