

Agrupamiento conceptual lógico combinatorio: una alternativa para la toma de decisiones

Yunia Reyes González, Alfonso Claro Arceo, Natalia Martínez Sánchez, Antonio Hernández Domínguez

Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km. 2^{1/2}. Torrens, municipio La Lisa. La Habana Cuba.

{yrglez, aclaro, natalia, ahdomínguez}@uci.cu

Abstract: Solving a problem leads to a process of identifying and selecting the appropriate action for their solution. This process is called Decision Making, where a decision is a choice between alternatives. The basis of the decision-making process is the information we have of the application domain. While more and better information you have, the greater quality in the definition of the problem, proposed solutions, the analysis of variants and selection of the most appropriate action will be obtained.

The motivation for this research lies in the use of the main benefit of logical combinatorial conceptual clustering algorithms, of providing to groupings with an explanation beyond the similarity between objects and groups of objects, as part of making process decisions.

Given that Based Systems Cases contribute to decision making, this proposal is described implementing its components: the knowledge base or base of cases, the module of cases recovery and the module of adaptation of solutions using LC-conceptual algorithm.

Keywords: Conceptual clustering, logical combinatorial pattern recognition, decision making, case-based systems.

Resumen Resolver un problema conduce a un proceso de identificación y selección de la acción adecuada para su solución. A este proceso se le denomina Toma de Decisiones, donde una decisión es la elección de una entre diversas alternativas. La base del proceso de toma de decisiones es la información que se tiene del dominio de aplicación. A más y mejor información, mayor calidad en la definición del problema, en las propuestas de solución, en el análisis de variantes y en la selección de la acción más conveniente.

La motivación de esta investigación radica en la utilización de la ventaja principal de los algoritmos de agrupamiento conceptual lógico combinatorio, de dar a los agrupamientos una explicación más allá de la similitud entre los objetos y las agrupaciones de objetos, en el proceso de toma de decisiones.

A partir de que los Sistemas Basados en Casos son una de las tecnologías actuales para la toma de decisiones se describe la propuesta implementando sus componentes fundamentales; la base de conocimiento o base de casos, el módulo de recuperación de casos y el módulo de adaptación de las soluciones utilizando el algoritmo LC-conceptual.

Palabras clave: Agrupamiento conceptual, Reconocimiento lógico combinatorio de patrones, Toma de decisiones, Sistemas basados en casos.

1 Introducción

Actualmente constituye un reto para los investigadores en el campo de la Inteligencia Artificial, aplicar esta ciencia a la amplia gama de problemas, cada vez creciente, que requieren su utilización. Se necesitan modelos computacionales que garanticen soluciones efectivas a partir de su validez y confiabilidad. En muchos campos de aplicación, es preciso además, que el modelo computacional que resuelve el problema sea comprensible para el usuario, y que las soluciones también puedan ser interpretadas por éste. Esto lleva a que continuamente se

propongan nuevos modelos teóricos; y se requieran además de herramientas computacionales que brinden facilidades a los investigadores de este campo para la validación y caracterización de los nuevos algoritmos, así como su aplicación concreta a la solución de un problema real [1][2][3].

Los Sistemas Basados en Casos (SBC) [4], [5] apoyan sus predicciones en ejemplos (casos) que se almacenan en la fase de aprendizaje. Una función de distancia o de semejanza determina los casos más semejantes al nuevo problema y las soluciones de los casos recuperados se adaptan para obtener una solución. Por tanto se puede afirmar que el proceso de adaptación está estrechamente vinculado con la toma de decisiones en un SBC.

En los últimos años, las investigaciones en el campo del Razonamiento Basado en Casos se han dirigido a la búsqueda de la eficiencia en la recuperación de los casos, donde se han desarrollado distintas técnicas que facilitan un acceso rápido a los casos relevantes de determinadas aplicaciones en grandes bases de casos. Para resolver este problema un número de métodos se han propuesto, incluyendo esquemas indizados. Lo esencial para el trabajo de un Sistema Basado en Casos es que todos los casos relevantes al nuevo problema puedan ser recuperados desde la base de casos eficientemente. En esto intervienen dos aspectos principales del sistema: cómo están organizados los casos en la base y cómo se realiza la comparación entre las descripciones del caso y el problema a resolver. Por eso, se dice que la recuperación tiene dos momentos principales: el acceso a los casos y la selección de los casos similares [2].

El Reconocimiento Lógico Combinatorio de Patrones (RLCP) [6], según la bibliografía consultada permite apreciar que puede ser aplicado para resolver problemas de la mayoría de las áreas del saber: reconocimiento de caracteres, diagnóstico médico [7], teledetección de la tierra, identificación de rostros humanos y huellas digitales, pronóstico de roturas en equipos y maquinarias, análisis de señales e imágenes biomédicas, inspección automática, conteo de células sanguíneas, análisis de los registros de pozos, arqueología, pronóstico de depósito de minerales, análisis de la actividad sísmológica, clasificación de documentos, entre otras. Sin embargo, no se encontraron trabajos que referencien la utilización de este enfoque en la toma de decisiones utilizando el paradigma del razonamiento basado en casos.

La investigación realizada propone un modelo para la elaboración de Sistemas Basados en Casos utilizando algoritmos de agrupamiento conceptual, en el que la estructura de representación del conocimiento garantiza un acceso y recuperación eficientes y permite una mayor fiabilidad en la determinación de nuevas soluciones.

2 Referentes teóricos sobre los algoritmos conceptuales

El agrupamiento conceptual no sólo propone obtener, además de los grupos estructurados a partir de una colección de objetos, el significado de esas agrupaciones, sino aportar información sobre el sentido que tiene que los objetos pertenezcan a un mismo grupo, brindar las características o conceptos para esas agrupaciones, definidas sobre la base de los rasgos en términos de los cuales se describen a esos objetos que conforman los grupos. La motivación esencial es lograr que el agrupamiento obtenido brinde al especialista del área específica una información acerca del significado como conjunto del agrupamiento.

El aprendizaje no supervisado se basa en las ideas de la Teoría Clásica de Conjuntos que plantea que los conjuntos no se definen, sino que se determinan y que existen dos formas para determinar un conjunto: de manera extensional, que significa dar los elementos que lo constituyen, y de manera intencional, que significa dar las propiedades que caracterizan al conjunto, que satisfacen los elementos que pertenecen al mismo [8], [6].

Los algoritmos clásicos de agrupamiento dan una estructuración extensional de los espacios y utilizan el primer procedimiento para dar los objetos que forman cada grupo. Estos algoritmos de agrupamiento, convencionales o tradicionales como también son conocidos en la literatura, presentan importantes limitaciones [9]: (I) Dejan el problema de la interpretación de los grupos al analista de datos. (II) No tienen en cuenta los métodos que los humanos emplean para agrupar objetos. Las personas tienden a agrupar objetos en categorías caracterizadas por conceptos, en grupos similares teniendo en cuenta algún atributo relevante o más importante que el resto. (III) Los métodos tradicionales de agrupamiento no toman en consideración ningún concepto o construcciones lingüísticas que las personas usan para describir colecciones de objetos.

Para enfrentar estas limitaciones a finales de los años 70 e inicios de los 80, Ryszard S. Michalski introdujo un conjunto de ideas que dieron origen al agrupamiento conceptual [10].

Los algoritmos de agrupamiento conceptual se componen de dos tareas fundamentales, que no tienen necesariamente que ser independientes ni realizarse en un orden determinado:

1. La estructuración o determinación extensional: es la enumeración de los objetos que componen los grupos, es donde se lleva a cabo el proceso de agrupar entidades en el que se determinan grupos a partir de una colección de objetos.
2. La caracterización o determinación intencional: es donde se determina el concepto de cada grupo de la estructuración, o sea, se generan las propiedades que caracterizan el agrupamiento.

Los algoritmos conceptuales se pueden dividir básicamente en dos clasificaciones: los incrementales y los no incrementales. Los primeros algoritmos desarrollados de tipo no incremental fueron los propuestos por Michalski CLUSTER/PAF [10], CLUSTER/2 [11], CLUSTER/S [12], más adelante surgieron otros como WITT [13], K-Means conceptual [14], LC-Conceptual [15], RGC [16] y CLUSTER/3 [17]. Entre los algoritmos de tipo incremental más representativos, se encuentran COBWEB [18], UNIMEM y RESEARCHER [19], EPAM [20], CLASSIT [21] y LINNEO+ [22], Galois [23], COBWEB/3 [24].

En el trabajo doctoral [6] se realiza un análisis crítico sobre los algoritmos de agrupamiento conceptual en función de las características y el funcionamiento de cada uno. La principal desventaja de los algoritmos conceptuales de tipo incremental es la dependencia del resultado (la estructuración) en función del orden de presentación de los objetos al algoritmo. Además de que cada algoritmo tiene otras limitaciones propias del funcionamiento o estrategia de solución, destacando como más importante desde el punto de vista de las aplicaciones, que en el caso de los algoritmos EPAM y COBWEB sólo manejan variables cualitativas. Por su parte CLASSIT únicamente maneja atributos cuantitativos y los algoritmos COBWEB/3 y LINNEO+ manejan información mezclada. Las descripciones generadas por el algoritmo COBWEB/3 están basadas en probabilidades, medias y varianzas, lo que resulta complicado de interpretar sobre todo para usuarios que no estén familiarizados con conceptos de probabilidad y estadísticas. Este algoritmo clasifica el nuevo objeto en los agrupamientos existentes y calcula un índice de calidad de la estructuración, para lo cual, analiza por separado los atributos cuantitativos y cualitativos y evalúa la contribución de cada uno de estos conjuntos de atributos por separado en la calidad de la estructuración. LINNEO+, por su parte, para el manejo de datos mezclados se basa en una distancia para el cálculo del parecido entre objetos y prototipos de los agrupamientos. En este caso, también se realiza por separado el cálculo de la distancia de los rasgos cuantitativos y cualitativos. Se considera la suma de estas dos distancias parciales como la distancia entre dos objetos en el espacio n-dimensional original [6].

En la línea no incremental, los algoritmos conceptuales CLUSTER/PAF, CLUSTER/2, CLUSTER/S y k-MEANS conceptual tienen como característica principal el hecho de requerir semillas para su funcionamiento, el número de semillas iniciales determina el número de agrupamientos. Este dato en la mayoría de los problemas prácticos es una incógnita, por lo que estos algoritmos en problemas donde no se conozca el número de agrupamientos a formar no resuelven el problema de manera satisfactoria. En el caso del algoritmo k-MEANS conceptual el cálculo del parecido entre dos objetos también se realiza calculando la distancia entre los atributos cuantitativos y cualitativos por separado, y posteriormente sumando ambos resultados para obtener la distancia total (considerando todos los atributos). Esta suma se considera como la distancia entre los objetos en el espacio n-dimensional original para el cálculo de centroides, esta distancia carece de sentido en el espacio n-dimensional original [6].

En [15] se reconoce que en la problemática de los datos mezclados, abordada por los algoritmos conceptuales incrementales y no incrementales se han propuesto las siguientes variantes:

1.-Codificar valores de los rasgos cualitativos como valores enteros numéricos, y aplicar medidas de distancia usadas en casos numéricos. El cambio de datos cualitativos a numéricos no tiene sentido, los códigos no son números, por lo que los valores de similaridad no son interpretables.

2.-Discretizar rasgos numéricos y aplicar algoritmos que manejen sólo información cualitativa. El proceso de discretización generalmente causa pérdida de información importante, especialmente la referente a la diferencia relativa ente valores para los rasgos numéricos. Anteriormente se han mencionado las dificultades de este paradigma así como la existencia de rasgos que no son discretizables. Además, de que el problema original tiene que ser modificado o cambiado de espacio de representación.

3.-Generalizar funciones de comparación diseñadas para rasgos cuantitativos para manejar rasgos cuantitativos y cualitativos. Las funciones usadas para rasgos cuantitativos están basadas en distancias que no pueden ser extendidas para manejar rasgos cualitativos por estar estos en un espacio diferente. Varios intentos o propuestas violan este hecho al calcular la distancia total como la suma de las distancias de los rasgos cuantitativos más la distancia de los rasgos cualitativos, y asumir que el resultado está en el espacio n-dimensional original donde el cálculo de centroides, por ejemplo, tendría sentido [15].

Una primera solución a esta problemática lo constituyó el algoritmo LC-Conceptual, de ahí su selección en esta investigación como propuesta de integración a los Sistemas Basados en Casos y además debido a cuatro aspectos fundamentales:

1. El hecho de ser un algoritmo cuya respuesta no es dependiente del orden de presentación de los objetos permite que no se condicione ninguna estructuración en función del orden de los objetos.
2. Tiene la capacidad de manejar atributos cualitativos y cuantitativos mezclados y con ausencia de información. Esta característica es significativa pues se presenta con frecuencia en las bases de casos construidas a partir de las bases de datos.

3. No requiere especificar el número de agrupamientos a priori ni semillas, lo que implica una gran ventaja pues es deseable organizar la base de casos en agrupaciones considerando las semejanzas entre los casos sin imponer una determinada cantidad de grupos a formar.
4. Los conceptos que genera son fáciles de interpretar por los usuarios pues se obtienen en función de las propias variables que describen los objetos y distinguen a los objetos de cada grupo. Esto facilita comprender las características fundamentales que describen las agrupaciones de casos y además reduce el espacio de búsqueda de los casos más similares dado un nuevo problema pues bastaría con comparar con los conceptos distintivos de cada grupo para determinar a cuál se parece más.

3 Descripción del algoritmo LC-Conceptual

El algoritmo LC-Conceptual [15], [25], [6] pertenece al conjunto de algoritmos de aprendizaje no supervisado basados en el enfoque lógico-combinatorio y retoma algunas ideas propuestas por Michalski para generar conceptos interpretables por los especialistas en términos del conjunto de rasgos original. Este algoritmo puede trabajar con atributos cualitativos y cuantitativos mezclados con ausencia de información, no requiere de semillas o especificar el número de agrupamientos a priori. Los agrupamientos se crean sobre la base de ciertas propiedades de la semejanza entre objetos y no sobre la base de criterios estadísticos o probabilísticos, los conceptos construidos no son descripciones estadísticas de los agrupamientos, que son difíciles de interpretar por los especialistas, sino propiedades lógicas basadas en los rasgos, en términos de los cuales se describen a los objetos en estudio; así también, cada agrupamiento tiene asociado uno o más conceptos que lo representan, tantos como testores típicos sean seleccionados. Sin embargo, posee una dificultad como todos los que empleen el uso de los testores típicos, asociado a la complejidad computacional en el cálculo de los mismos.

El algoritmo LC consta de dos etapas: una de estructuración extensional, donde se forman los agrupamientos y otra de estructuración intencional, donde se caracteriza cada agrupamiento mediante una propiedad lógica o concepto.

En la primera etapa de estructuración extensional se utilizan los conceptos del enfoque lógico combinatorio para un problema no supervisado. En ella se construyen los agrupamientos de objetos basándose en la semejanza entre ellos y se utiliza un criterio de agrupamiento. Los objetos pueden ser representados en una matriz inicial (MI), $MI = \{I(O_1), I(O_2), \dots, I(O_n)\}$ un conjunto de descripciones de los objetos O_1, O_2, \dots, O_n de un universo U , dadas como $I(O_i) = (x_1(O_i), \dots, x_n(O_i))$. Para cada x_i se tiene asociado un conjunto de valores admisibles M_i $i=1, \dots, n$, donde el espacio de representación inicial de los objetos, $M_1 \times M_2 \times \dots \times M_n$ es el producto cartesiano de los conjuntos admisibles de valores de los rasgos x_1, \dots, x_n . Para cada rasgo x_i ($i=1, \dots, n$), se asocia un criterio de comparación C_i : $M_i \times M_i \rightarrow L_i$, donde C_i es una evaluación del grado de semejanza entre cualquier par de valores de la variable x_i cuando L_i es un conjunto totalmente ordenado, que alcanza valor máximo 1, cuando existe la mayor similitud entre los rasgos que se comparan y valor mínimo 0 cuando la similitud entre los rasgos es la menor.

Entre las descripciones de objetos se define una función de semejanza, la cual determina una evaluación del grado de similitud de un objeto con respecto al otro teniendo en cuenta las similitudes entre los rasgos: $\beta: (M_1 \times \dots \times M_n)^2 \rightarrow L$, siendo L un conjunto totalmente ordenado.

A partir de MI y β se puede construir una matriz que refleje las relaciones de semejanza entre todos los objetos sujetos a estudio. A esta matriz se le llama matriz de semejanza (MS) y es: $MS = \|\beta(I(O_i), I(O_j))\|_{m \times n}$. La matriz de semejanza es simétrica si lo es la función de semejanza. Para determinar la similitud entre rasgos, es importante considerar la naturaleza de los mismos, en dependencia de esta, se utilizan diferentes criterios de comparación [26].

La magnitud $\beta_0 \in [0, 1]$ se denomina umbral de semejanza, y es la cota inferior de los valores de semejanza existentes entre los objetos de la MS. Este parámetro es necesario para conformar los grupos de la estructuración y puede ser dado por expertos del área de aplicación o calculado de alguna de las siguientes maneras [27]:

$$a) \beta_0 = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \beta(I(O_i), I(O_j)).$$

$$b) \beta_0 = \frac{1}{m} \left\{ \sum_{i=1}^m \max_{j=1, \dots, m} \{\beta(I(O_i), I(O_j))\} \right\}.$$

$$c) \beta_0 = \min_{i=1, \dots, m-1} \left\{ \min_{j=i+1, \dots, m} \{\beta(I(O_i), I(O_j))\} \right\}.$$

El criterio de agrupamiento Π [6], unido a la función de semejanza y a la existencia de otros objetos, es la razón por la cual un objeto va a pertenecer a un agrupamiento o el por qué dos objetos pertenecerán a una misma

agrupación. De esta manera se puede apreciar que la selección del criterio a usar, es determinante en la calidad de la solución del problema no supervisado.

La definición del criterio de agrupamiento debe estar basada en el conocimiento que se tenga sobre el problema en concreto que se está tratando, para poder definir así el tipo de comportamiento entre los objetos a partir de sus semejanzas, que resulte significativo, según el problema en particular. Por tanto al seleccionar algún criterio de agrupamiento, dado un conjunto de objetos y la función de semejanza, se ha definido indirectamente, la familia de agrupaciones, es decir, la estructura del universo ha sido conformada.

El planteamiento formal de la estructuración de universos, en el aprendizaje no supervisado, consiste en encontrar un criterio de agrupamiento que responda a los intereses del problema en cuestión.

Dos descripciones (objetos) $I(O_n), I(O_j)$ se denominan β_0 -semejantes si $\beta(I(O_i), I(O_j)) \geq \beta_0$. Sea un espacio de representación $\Phi = \{\Xi, \beta\}$, sea dado un conjunto de descripciones de objetos $MI = \{I(O_1), I(O_2), \dots, I(O_m)\}$, una función de semejanza β y un umbral β_0 :

Criterio agrupacional β_0 Compacto [6]

Se dice que $B \subseteq MI$, $B \neq \emptyset$ es un conjunto β_0 -compacto si y sólo si:

- Todo elemento de B tiene en B al elemento que más se le parece que es β_0 -semejante con él.
- No existe fuera de B un elemento cuyo elemento más parecido que sea β_0 -semejante esté en B.
- B debe ser el conjunto más pequeño de cardinalidad mayor que 1.
- Todo elemento β_0 -aislado constituye un conjunto β_0 -compacto (*degenerado*).

Criterio agrupacional β_0 -conexo [6]

Se dice que $C \subseteq MI$, $C \neq \emptyset$ es una componente β_0 -conexa si y sólo si:

- Para cualquier par de elementos de C existe una sucesión de elementos en C, que empieza en O_i y termina en O_j tales que uno es β_0 -semejante al siguiente.
- No existe fuera de C un elemento β_0 -semejante a un elemento de C.
- Todo elemento β_0 -aislado es una componente β_0 -conexa (*degenerada*).

Cada uno de los conjuntos anteriormente generados constituye un agrupamiento y con ellos se estructura la Matriz de Aprendizaje, la cual contiene los objetos con la información del agrupamiento al cual pertenecen.

En la segunda etapa de estructuración intencional o conceptual se construyen las propiedades (conceptos) que caracterizan a cada agrupamiento de objetos utilizando, para ello, el operador de Refunión Condicionada y los testores típicos.

En esencia, un testor es un conjunto de características (rasgos) que diferencian a elementos (objetos) de clases distintas. Los testores típicos constituyen variantes minimales de subconjuntos de rasgos, existiendo algoritmos que permiten calcular la importancia del rasgo a partir de estos [15]. Los testores típicos se obtienen como resultado de aplicar un algoritmo que admite como entrada una matriz básica. Estos testores típicos se utilizan para la construcción de los l-complejos resultantes de aplicar el operador Refunión Condicionada [6].

Un selector es una proposición relacional $[x_i \# R_i]$, donde R_i es un conjunto de valores del rasgo x_i y el símbolo $\#$ denota un operador relacional. Para el caso de los rasgos no numéricos éstos son $\{=, \neq, \in, \notin\}$ y para los numéricos $\{>, <, \leq, \geq, \in, \notin\}$.

Un producto lógico de selectores, se denomina complejo lógico o l-complejo. Se dice que un objeto satisface un l-complejo (o un l-complejo cubre a O) si los valores de los rasgos en O satisfacen todos los selectores del l-complejo.

Ya construidos los agrupamientos se caracteriza cada uno de éstos utilizando para ello el operador de Refunión Condicionada (RUC). Este operador utiliza al conjunto de los testores típicos para construir propiedades o conceptos que satisfacen los objetos de cada agrupamiento y por cada testor típico se obtiene un l-complejo. La importancia de utilizar los testores típicos radica en que estos no contienen todos los rasgos, sino los más relevantes al problema en cuestión, por tanto cuando se haga referencia al conjunto de rasgos, debe entenderse que son sólo aquellos rasgos presentes en el testor típico con que se trabaje.

El operador de Refunión Condicionada transforma un conjunto de objetos y/o l-complejos en un conjunto de l-complejos, y se define de la siguiente manera: RUC: $2^U \cup 2^{L(U)} \rightarrow 2^{L(U)}$. Se determina para cada variable el conjunto de valores que la variable toma pero no de manera independiente, sino en combinación con el resto de las variables [6]. Estos serán los conceptos que describen los agrupamientos formados.

Para aplicar el procedimiento RUC se parte de la matriz de aprendizaje y los testores típicos. Se toma el primer objeto del primer agrupamiento y se construyen conjuntos (R_1, \dots, R_p) con los valores de cada rasgo del objeto, por tanto se obtendrán p conjuntos de valores de rasgos, donde p es la cantidad de rasgos que tiene el objeto. Luego se considera el segundo objeto, el valor del primer rasgo se incorpora al subconjunto R_1 , si sólo sí, ese

valor en combinación con los valores de los conjuntos de rasgos R_2, \dots, R_p , no está presente en ninguno de los objetos de los demás agrupamientos. Esta condición debe irse comprobando para cada valor de cada rasgo del segundo objeto y esa es la manera de ir construyendo los l-complejos (conceptos) para el primer agrupamiento, pues el proceso se repite con todos los objetos del grupo. De igual forma, se generan los conceptos para el resto de los agrupamientos. Estos conceptos cumplen la propiedad que no existe fuera del agrupamiento, un objeto descrito por ellos.

En el algoritmo LC-Conceptual los agrupamientos generados no son necesariamente particiones. Se obtendrán particiones o cubrimientos en dependencia del criterio agrupacional Π seleccionado. Una dificultad que posee es que, al aplicar el operador Refunción Condicionada a cada agrupamiento K_i , pueden quedar objetos de K_i no cubiertos por el concepto construido. Este algoritmo tiene, además, una complejidad computacional exponencial en el peor de los casos destacándose, en esto, el cálculo de los testores típicos y el proceso de generación de todas las combinaciones en la aplicación de la Refunción Condicionada.

4 Modelo conceptual para la toma de decisiones en los Sistemas Basados en el Conocimiento

Los sistemas basados en el conocimiento [2] son una rama de la Inteligencia Artificial útil para enfrentar la construcción de Sistemas de Apoyo a la Toma de Decisiones, dado que comparten aspectos afines. Estos sistemas utilizan conocimiento sobre un dominio específico y la solución que se obtiene es similar a la obtenida por una persona experimentada en el dominio del problema.

Una característica distintiva de los sistemas basados en el conocimiento es la separación del conocimiento (base de conocimiento) del método de solución del problema (máquina de inferencia). La construcción de la base de conocimiento lleva implícito un arduo proceso de adquisición del mismo y es particular para cada sistema, por lo que será necesario construirla para cada aplicación. Sin embargo, la máquina de inferencia puede reutilizarse en la construcción de sistemas basados en el conocimiento, siempre que el tipo de conocimiento y el tipo del razonamiento sea similar.

Diferentes tipos de conocimiento dan lugar a diferentes tipos de sistemas basados en el conocimiento, entre ellos los sistemas basados en reglas, basados en probabilidades, sistemas expertos conexionistas o redes expertas y los sistemas basados en casos [2].

La máquina de inferencia es el método implementado que utiliza el conocimiento de la base para resolver los problemas del dominio. El tipo de conocimiento determina qué método de solución de problemas es posible utilizar.

4.1. Modelo para diseñar Sistemas Basados en Casos utilizando el agrupamiento conceptual

El Razonamiento Basado en Casos (RBC) [2], [3] es un enfoque que aborda nuevos problemas tomando como referencia problemas similares resueltos en el pasado. De modo que problemas similares tienen soluciones similares, y la similitud juega un rol esencial. Sus componentes fundamentales son la base de casos, el módulo de recuperación de casos y el módulo de adaptación de las soluciones [28].

La base de casos contiene las experiencias, ejemplos o casos a partir de los cuales el sistema hace sus inferencias. Esta base puede ser generada a partir de casos o ejemplos resultantes del trabajo de expertos humanos o por un procedimiento semiautomático-automático, que construye los casos desde datos existentes registrados, por ejemplo, en una base de datos.

En el módulo de acceso y recuperación es donde se recuperan de la base de casos los casos más semejantes al problema y posteriormente la determinación de los casos más semejantes, de esta manera las soluciones contenidas en dichos casos pueden utilizarse directamente como solución al nuevo problema, pero comúnmente necesitan ser modificadas.

Un Razonador o Sistema Basado en Casos (SBC) depende de la estructura y el contenido de la base de conocimiento. Hasta el presente, para resolver el problema de organizar una base de casos, un enfoque ha sido almacenar los casos de forma secuencial y analizarlos todos para resolver el nuevo problema. Este tipo de organización hace lento el proceso de recuperación. Un método alternativo consiste en particionar los casos en grupos y organizarlos jerárquicamente.

El modelo (figura 1) que se describe consta de dos fases y se fundamenta en la idea esencial del algoritmo LC-Conceptual.

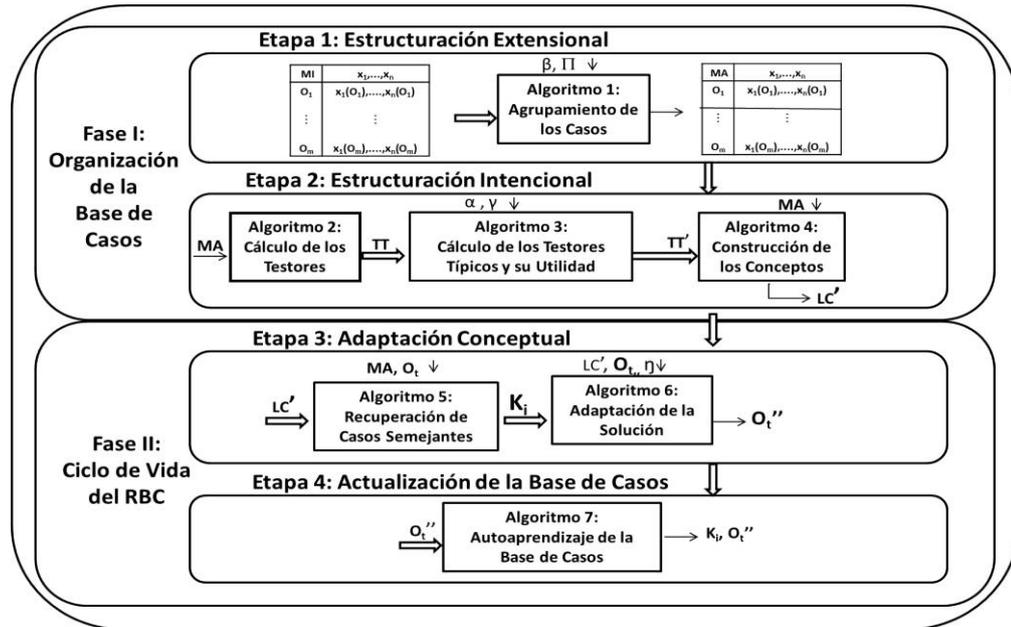


Figura 1. Modelo para la toma de decisiones en un SBC (Fuente: Elaboración propia)

Cada una de las fases se compone de dos etapas con un nivel de precedencia establecido. Las etapas se sustentan en siete algoritmos basados en el reconocimiento lógico combinatorio de patrones.

La primera fase denominada “Organización de la Base de Casos”, se corresponde con la ingeniería del conocimiento que es necesaria para desarrollar sistemas basados en casos. Este proceso comprende la organización de la base de casos y consta de dos etapas: una primera donde se estructuran los grupos a formar a partir de las similitudes entre los casos y la siguiente donde se generan los conceptos para dichas agrupaciones.

La segunda fase “Ciclo de vida del RBC”, implementa el método de solución de problemas de los SBC. Se divide a su vez en dos etapas: la primera denominada Adaptación Conceptual, la cual comprende el acceso y recuperación de los casos más semejantes y la aplicación de un nuevo procedimiento para la adaptación de las soluciones teniendo en cuenta los conceptos que describen los casos similares rescatados de la BC y una segunda denominada Actualización de la Base de Casos, etapa final que cierra el ciclo del SBC, es aquí donde se incorpora a la memoria de casos, el nuevo caso con los valores correspondientes en los rasgos objetivos.

FASE I Organización de la base de Casos

Etapa 1: “Estructuración extensional utilizando el enfoque lógico combinatorio para problemas no supervisados”.

Se describen los algoritmos que sustentan la propuesta, tomando como dato de entrada principal la matriz inicial MI , como ilustra la figura 2, donde n es la cantidad de rasgos (características que describen los casos) y m el número de objetos (casos),

	x_1	x_2	...	x_n
O_1	$x_1(O_1)$	$x_2(O_1)$...	$x_n(O_1)$
\vdots			\vdots	
O_m	$x_1(O_m)$	$x_2(O_m)$...	$x_n(O_m)$

Figura 2. Matriz Inicial.

Algoritmo 1. Agrupamiento de los casos

Entrada: MI, β, Π

Salida: MA //matriz estructurada en c agrupamientos (K_i).

Paso 1: Construir la Matriz de Semejanza utilizando una función de semejanza β .

Paso 2: Calcular el umbral de semejanza utilizando un criterio β_0 .

Paso 3: Agrupar siguiendo un criterio agrupacional Π .

Como dato de salida del algoritmo 1 se obtiene la matriz de aprendizaje (MA), la cual está estructurada en agrupamientos de casos β_0 -semejantes, culminando así la primera etapa con la estructuración extensional.

La figura 3 ilustra un ejemplo de las diferentes matrices que se generan como resultado de aplicar el algoritmo 1, usando como criterio de comparación por rasgos la igualdad estricta y como función de semejanza la suma normalizada de los rasgos.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1	1	0	1	0	1	0	1	0	1
0	0	1	0	1	0	1	0	1	0
1	1	0	1	0	1	1	0	1	0
1	0	0	0	0	1	1	1	1	0
0	1	1	1	0	1	0	0	1	0

Matriz Inicial formada por 5 objetos y 10 rasgos binarios

Matriz Semejanza					
Objetos	O1	O2	O3	O4	O5
O1	1.0	0.0	0.6	0.5	0.5
O2	0.0	1.0	0.4	0.5	0.5
O3	0.6	0.4	1.0	0.7	0.7
O4	0.5	0.5	0.7	1.0	0.4
O5	0.5	0.5	0.7	0.4	1.0

MS formada a partir de la MI con $\beta_0=0.6$ y criterio agrupacional β_0 -Compacto

Matriz de Aprendizaje										
X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Class
1	1	0	1	0	1	0	1	0	1	1
0	0	1	0	1	0	1	0	1	0	2
1	1	0	1	0	1	1	0	1	0	1
1	0	0	0	0	1	1	1	1	0	1
0	1	1	1	0	1	0	0	1	0	1

Matriz de Aprendizaje con la pertenencia de los objetos a las clases

Figura 3. Matrices generadas como resultado de la ejecución del algoritmo 1.

Etapa 2: Estructuración intencional utilizando el cálculo de los testores

Para pasar a la etapa 2 es necesario calcular los testores. De este modo se reduce eficientemente la cantidad de rasgos con los cuales se deben describir los objetos y se encuentran los rasgos que inciden de manera determinante en un problema, utilizando como dato de entrada la matriz de aprendizaje (MA).

En el algoritmo 2 primero se calcula la Matriz de Diferencias (MD) y luego la Matriz Básica (MB), a partir de la cual se determinan los testores. La matriz de diferencias es una matriz booleana que se obtiene de la matriz de aprendizaje, comparando los respectivos valores de los rasgos en objetos de clases diferentes, por medio de los criterios de comparación de valores de las variables. El resultado de estos criterios es binario (independientemente del tipo de rasgo), será 1 cuanto más diferentes sean los rasgos y 0 mientras más semejantes, los mismos deben ser construidos por el experto del dominio de aplicación. Dada una matriz de diferencias, se puede obtener la matriz básica, formada exclusivamente por filas básicas de MD.

La fila i_t es básica sí y sólo sí en MD no existe fila i_p alguna que sea subfila de i_t [26].

Sea i_p, i_t filas de MD . Se dice que i_p es subfila de i_t sí y sólo sí:

- $\forall j (a_{ij}=0 \Rightarrow a_{ipj}=0)$
- $\exists j_0 (a_{ij_0}=1 \wedge a_{ipj_0}=0)$

Las filas de la matriz básica constituyen el conjunto de testores.

Algoritmo 2. Cálculo de los testores

Entrada: MA

Salida: TT //Conjunto de testores.

Paso 1. Calcular la MD

Paso 2. Calcular la MB // las filas de la MB conforman el conjunto TT de testores.

La figura 4 muestra las matrices de diferencias y básica calculadas para el ejemplo anterior.

Matriz de Diferencias										
X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	
1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	0	0	0	0	
1	0	1	0	1	1	0	1	0	0	
0	1	0	1	1	1	1	0	0	0	

Matriz Básica										
X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	
1	1	1	1	1	1	0	0	0	0	
1	0	1	0	1	1	0	1	0	0	
0	1	0	1	1	1	1	0	0	0	

Figura 4. Matrices generadas como resultado de la ejecución del algoritmo 2.

A partir de los testores resultantes de la matriz básica calculada en el algoritmo 2, deben calcularse los testores típicos, para ello se utiliza el algoritmo LEX. Como el número de testores típicos puede ser alto, cada agrupamiento podría tener asociado una gran cantidad de conceptos o propiedades. En el algoritmo 3 además de calcular los testores típicos, se propone un criterio para el cálculo de una medida, que los autores denominan utilidad del testor. Esta medida se tiene en cuenta para la selección de los testores que se utilizan para construir los conceptos. El número de testores típicos seleccionados está estrechamente relacionado con el criterio del experto según el problema a resolver.

Algoritmo 3. Cálculo de los testores típicos y su utilidad

Entrada: TT, α y γ

Salida: TT' // Conjunto de testores típicos de mayor utilidad.

Paso 1. Calcular los testores típicos utilizando el algoritmo LEX[29].

Paso 2. Calcular el peso ε_i de los rasgos x_i que aparecen en la familia de testores típicos según: $\varepsilon(x_i) = \alpha F(x_i) + \gamma L(x_i)$ donde: $\alpha > 0$, $\gamma > 0$ y $\alpha + \gamma = 1$.

// α y γ son dos parámetros determinados por el experto que ponderan la participación o influencia de F_i (frecuencia de aparición) y L_i (longitud de los testores) respectivamente.

$$F(x_i) = \frac{|TT(x_i)|}{|TT|} \quad L(x_i) = \frac{\sum_{t \in TT(x_i)} \frac{1}{|t|}}{|TT(x_i)|}$$

$|TT(x_i)|$ número de testores donde aparece el rasgo x_i .

$|TT|$ número de testores.

$|t|$ número de rasgos que forman el testor t .

Paso 3: Selección del conjunto de rasgos relevantes:

a. Para cada testor t_i calcular: $\psi(t_i) = \sum_{x \in t_i} \varepsilon(x)$

//la magnitud calculada es la suma de la importancia de los rasgos que componen el testor.

Paso 4. Seleccionar los q -testores de mayor $\psi(t_i)$.

//donde q es un parámetro relacionado con el problema a resolver y TT' es el conjunto de los q testores de mayor $\psi(t_i)$.

Siguiendo la idea básica del algoritmo LC-Conceptual se aplica el algoritmo 4 para la estructuración intencional o conceptual, donde se construyen las propiedades (conceptos) que caracterizan a cada agrupamiento de objetos utilizando los testores típicos.

Algoritmo 4. Construcción de los conceptos

Entrada: MA, TT'

Salida: LC' //Conjunto de los conceptos para cada K_i .

Paso 1. Para cada agrupamiento K_i , $i = 1, \dots, c$, calcular los l -complejos aplicando el operador RUC.

// Los conceptos que caracterizan a cada agrupamiento K_i serán los l -complejos obtenidos en el paso anterior.

Una vez aplicado el algoritmo 4 se obtienen los conceptos que se utilizan en el proceso de adaptación de la solución al nuevo problema y finaliza así la primera fase del modelo.

FASE II Ciclo de vida del RBC

El ciclo general del RBC se basa en que, dado un nuevo caso se accede y recuperan los m casos más semejantes y a partir de los mismos se toma la decisión de la solución, ya sea la solución de uno de los casos recuperados sin modificación o aplicando una de las técnicas de adaptación existentes. Culminada esta fase, la base de casos queda organizada jerárquicamente en agrupamientos que contienen los casos según su grado de semejanza y cada grupo está representado por los conceptos que lo distinguen del resto de los agrupamientos. Lo anterior, facilita el método de acceso y recuperación de los casos.

Etapas 3: Proceso de adaptación utilizando los conceptos construidos en cada grupo.

La tercera etapa se corresponde con el proceso de adaptación utilizando los conceptos construidos en cada grupo. Los algoritmos 5 y 6 describen el ciclo de vida del SBC, incluida la adaptación a la solución con el método que se propone en esta investigación.

Dado un nuevo caso, se aplica el algoritmo 5, que compara el nuevo caso con los conceptos de cada agrupamiento y selecciona a qué grupo pertenece; una vez seleccionado, se tienen los casos más semejantes al nuevo caso. Adicionalmente, en el proceso de implementación se obtienen los conceptos por agrupamientos, y una vez seleccionado el grupo se aplica el algoritmo 6, que se corresponde con el módulo de adaptación y se asigna la solución al nuevo caso. Es así como dado un caso nuevo para asignarle la solución que más se adapte a sus características, sólo se aplican los algoritmos 5 y 6 correspondientes al ciclo de vida del SBC.

Como para la selección del testor o testores de mayor utilidad, se calculó la importancia ($\epsilon(x_i)$) de todos los rasgos que conforman la familia de testores TT, esta importante información se tiene en cuenta al calcular la semejanza del nuevo caso (O_t) con los conceptos que representan cada clase.

Algoritmo 5. Recuperación de los casos más semejantes

Entrada: MA, LC' conceptos asociado a cada agrupamiento K y O_t (nuevo caso)

Salida: K_i seleccionado

Paso 1. Para cada K_i de MA calcular:

- a. $\beta(O_t, O_i)$ // donde O_i son los objetos que se corresponden con los LC'. Es válido aclarar que sólo se comparan los rasgos que conforman el LC'.
- b. Calcular:

$$\lambda_i(O_t) = \frac{\sum \beta(O_t, O_i)}{|I|} \quad // \text{ donde } O_t \text{ es el nuevo caso, } O_i \text{ los conceptos y}$$

$|I|$ el número de conceptos asociados a la clase i .

Paso 2.

- a. Seleccionar el $\max(\lambda_i(O_t))$
- b. Seleccionar el agrupamiento K_i // los objetos del grupo i se corresponden con los objetos más semejantes al nuevo objeto.

A. Adaptación nula conceptual

Hasta aquí se puede afirmar que la utilización del enfoque del agrupamiento conceptual implica un valor agregado a la tan utilizada adaptación nula, la cual consiste en recuperar los casos más semejantes al nuevo caso y dar toda la responsabilidad al especialista de tomar la decisión realizando un análisis a partir de los casos recuperados, no obstante, este enfoque va más allá, pues además de ofrecer al especialista los casos más semejantes, ofrece los conceptos distintivos de los mismos facilitándole así la toma de decisiones.

B. Adaptación transformacional por sustitución utilizando el agrupamiento conceptual

El algoritmo 6 propone un procedimiento para otro tipo de adaptación: la transformacional por sustitución utilizando los algoritmos de agrupamiento conceptual.

El algoritmo 6 calcula la semejanza del objeto O_t con los objetos de la clase que resultó seleccionada en el algoritmo 5. En el cálculo de la semejanza se tiene en cuenta la importancia (ϵ_i) calculada con anterioridad a partir de los rasgos que conforman los testores típicos. A todos los rasgos que no conforman los testores típicos, se le asigna una importancia menor que la mínima ϵ_i calculada. Cada l-complejo (LC') se sustituye en el nuevo caso (esto consiste en tomar el nuevo caso O_t y reemplazar los valores del l-complejo en él, transformándose en un nuevo caso O'_t). Luego se calcula la semejanza de ese nuevo caso con los valores sustituidos (O'_t) con cada uno de los casos recuperados. Se integra en una misma ecuación la semejanza calculada entre los casos de la clase con el nuevo caso y con el nuevo caso modificado. Para determinar la solución al nuevo caso se selecciona el caso con mayor valor de semejanza y se asigna el valor del rasgo objetivo de ese caso al rasgo objetivo del nuevo caso presentado.

Algoritmo 6. Adaptación de la solución

Entrada: K_i, η, LC' y O_t // O_t nuevo caso.

Salida: O_t'' // nuevo caso con la solución asignada aplicando la adaptación transformacional por sustitución utilizando el agrupamiento conceptual.

Paso 1. Calcular $\beta(O_t, O_j)$ // donde O_j son los objetos del agrupamiento K_i y se calcula la semejanza del nuevo objeto con cada uno de los objetos que conforman el grupo.

Paso 2. Para cada l-complejo (LC') sustituirlo en O_t :

a. Calcular $\beta'(O'_t, O_j)$

// O'_t es el objeto O_t resultante de sustituir los l-complejos y se calcula la semejanza del nuevo objeto modificado con todos los objetos del agrupamiento.

b. Calcular: $\Gamma_j(O_t) = \eta\beta(O_t, O_j) + (1-\eta)\beta'(O'_t, O_j)$

// $j=1 \dots |K|$ y η depende del problema, a medida que η tiende a 0 significa que se le está dando mayor importancia a la semejanza teniendo en cuenta los l-complejos.

c. Calcular $\Phi_r(O'_t, O_j) = \max(\Gamma_j(O_t))$

// r representa el l-complejo, en otras palabras terminado el paso 2 se tendrán tantos $\Phi_r(O'_t, O_j)$ como LC' tenga asociado el agrupamiento.

Paso 3. Seleccionar el mayor $\Phi_r(O'_t, O_j)$ y asignar la solución del objeto O_j al objeto O_t'' .

// se selecciona el objeto O_j cuyo grado de semejanza con el nuevo objeto es mayor, teniendo en cuenta ambas semejanzas calculadas en los pasos 1 y 2 respectivamente.

Por último, para cerrar el ciclo de vida del SBC, el nuevo caso puede ser incorporado a la base a través del proceso de actualización de la memoria de casos.

Algoritmo 7. Autoaprendizaje de la Base de Casos

Entrada: O_t'' // objeto modificado con el l-complejo del agrupamiento K_i que resultó con mayor $\Phi_r(O'_t, O_j)$

Salida: K_i con el objeto O_t'' asignado

Paso 1. Asignar el nuevo objeto O_t'' al agrupamiento K_i

4.2. Valoraciones sobre el modelo propuesto

El modelo propuesto puede parecer muy costoso teniendo en cuenta la complejidad computacional de los algoritmos del enfoque lógico combinatorio que se utilizan, tales como el algoritmo LEX para el cálculo de los testores típicos y el LC-Conceptual para el cálculo de los l-complejos, pero ambos sólo se utilizan durante la etapa de ingeniería del conocimiento implícita en la implementación de un Sistema Basado en Casos.

El método aporta a todo el proceso de implementación de un SBC, desde la ingeniería del conocimiento necesaria hasta al método de solución del problema (RBC) que se utiliza en este tipo de sistema basado en el conocimiento.

Como resultado de aplicar algoritmos del enfoque lógico combinatorio para el agrupamiento (no supervisado) y tomar los conceptos como representantes de cada grupo, se obtiene un modelo jerárquico de la base de casos que permite una búsqueda más eficiente.

Debe tenerse en cuenta en la estructuración resultante de aplicar el algoritmo 1, el problema de los grupos no balanceados, pues en el caso extremo de que se generen dos grupos uno con muy pocos ejemplos y otro que contiene a la mayoría, el sistema no se entrenaría de manera adecuada. En tal situación es recomendable el uso de técnicas para la edición de los conjuntos de entrenamiento ya bien sea reduciendo la clase con mayor cantidad de ejemplos, o ampliando la de menor cantidad. Una solución en este sentido puede ser mediante el empleo de la teoría de los conjuntos aproximados como se propone en [30].

5 Resultados Experimentales

Para analizar los resultados del modelo, siguiendo las experiencias de muchos investigadores en el campo de la Inteligencia Artificial, se utiliza la información disponible en el Uci Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets.html>), de ahí se tomaron siete bases de casos cuyas características se ilustran en la tabla 1.

Tabla1. Características de las Bases de Casos utilizadas en la experimentación

Bases de Casos	Cantidad de Casos	Rasgos Predictores	Rasgos Objetivos	Rasgos Numéricos	Rasgos No Numéricos	Ausencia de Información
Heart Disease (Statlog)	270	13	1	Sí	Sí	No
Breast-Cancer	286	9	1	No	Sí	Sí
Primary-Tumor	339	17	1	No	Sí	Sí
Hepatitis	155	19	1	Sí	Sí	Sí
Labor	57	16	1	Sí	Sí	No
Postoperative-Patient-Data	90	8	1	Sí	Sí	Sí
Zoo	101	17	1	Sí	Sí	No

Para determinar la eficiencia del modelo propuesto, se realiza una comparación de la eficiencia en el acceso a los casos más semejantes entre las base de casos organizadas con estructura plana y con la estructura jerárquica propuesta en la primera fase del modelo. Este valor de la eficiencia está dado por la medida en que es posible reducir el espacio de búsqueda para acceder a los casos más similares con el nuevo problema presentado.

Si se organizan las Bases de Casos con una estructura plana, se deben realizar como máximo, tantas comparaciones como número de casos existan hasta encontrar el caso más similar. Mientras que con la organización de la base de casos propuesta en el modelo, se realizan comparaciones sólo con los conceptos que representan cada grupo y se determina con cuál caso nuevo es más similar. Esto implica una reducción significativa del espacio de búsqueda al comparar únicamente con los conceptos que describen los grupos y solamente con los rasgos que componen esos conceptos, de ese modo también se reducen las comparaciones.

La reducción de la cantidad de comparaciones es un resultado bueno pero no suficiente, debe demostrarse además que la calidad de la respuesta no se afecta, para ello se analizaron los resultados en la recuperación de los casos más semejantes y de la efectividad en los casos seleccionados para la adaptación.

La medida seleccionada para evaluar la propuesta de Sistema Basado en Casos descrita en esta investigación, fue la precisión de la clasificación, entendiendo esta como el promedio de acierto en la clasificación de nuevos casos. Para llevar a cabo este proceso se empleó el método de validación cruzada (k-folds cross validation) usando 10 subconjuntos distribuidos uniformemente y se realizaron experimentos con las siete bases de casos de estudio. Se utilizó un subconjunto como test y el resto como training hasta completar las 10 iteraciones, calculándose el porcentaje de casos clasificados correctamente para cada una de las bases de casos.

El objetivo fue comparar los resultados obtenidos por un SBC estándar (sin previa organización de la base de casos y realizando adaptación nula) y los arrojados por el modelo de SBC aquí presentado (organización jerárquica de la base de casos y adaptación utilizando los conceptos). En la tabla 2 se muestran los valores de la precisión en la clasificación para las bases de casos, la tercera columna representa la diferencia entre la precisión de la clasificación realizada por un SBC estándar y el SBC propuesto.

Tabla 2. Resultados de la precisión en la clasificación

Bases de Casos	SBC propuesto	SBC estándar	Δ
Heart Disease (Statlog)	84,69%	84,03%	+ 0,66
Breast-Cancer	92,55%	91,85%	+ 0,70
Primary-Tumor	87,03%	85,93%	+ 1,10
Hepatitis	83,78%	82,97%	+ 0,81
Labor	86,50%	86,50%	0,00
Postoperative- Patient-Data	82,37%	82,26%	+ 0,11
Zoo	89,87%	88,92%	+ 0,95

Como puede apreciarse en la tabla 2, los resultados de la clasificación realizada por el SBC propuesto mejoran con respecto a un SBC estándar en seis de las bases de casos utilizadas y en una de ellas se igualan. Dadas las características de las bases de casos empleadas en los experimentos de presentar datos mezclados (numéricos y no numéricos) y de trabajar con valores ausentes y los resultados obtenidos, se demuestra que el modelo propuesto puede ser valorado como una alternativa válida para la construcción de SBC en dominios con similares particularidades.

Esta contribución forma parte de una investigación más amplia que se realiza actualmente, donde se pretende extender los experimentos incluyendo un mayor número de bases de casos con variadas características para comprobar su validez en diferentes dominios e introducir la construcción de este tipo de SBC en la resolución de interesantes problemas prácticos como el diagnóstico médico, la elaboración de sistemas tutoriales inteligentes, entre otros.

6 Conclusiones

En este artículo se describe una vía que facilita el proceso de adaptación en los SBC, ya sea complementando el método de adaptación nula (el más utilizado en la literatura científica revisada en este campo) o un nuevo método utilizando el agrupamiento conceptual, el que sobrepasó las expectativas de los autores, ya que aporta a todo el proceso de implementación de un SBC, desde la ingeniería del conocimiento necesaria hasta al método de solución del problema (razonamiento basado en casos) que se utiliza en este tipo de sistema basado en el conocimiento. A partir del objetivo que motivó esta investigación se pueden plantear las conclusiones:

1. Como resultado de aplicar algoritmos del enfoque lógico combinatorio para el agrupamiento no supervisado y tomar los conceptos como representantes de cada grupo, se obtiene un modelo jerárquico de la base de casos que permite una búsqueda más eficiente, ya que se reduce la complejidad temporal de $O(n)$ a un $O(\log n)$.
2. El cálculo de los testores y testores típicos para obtener los conceptos por grupo proporciona la selección de los rasgos predictores más relevantes y el cálculo de la importancia de los mismos, lo que influye favorablemente en la complejidad temporal, pues en la comparación entre casos no se tienen en cuenta aquellos rasgos irrelevantes o redundantes y sí se tiene en cuenta la importancia de los mismos, lo que hace más eficiente y eficaz la toma de decisiones.
3. Facilita la implementación del ciclo de vida de un SBC, pues dado un nuevo caso se compara con los conceptos representantes de cada grupo y se decide a qué grupo pertenece, siendo los casos que conforman dicho grupo los casos más semejantes al nuevo problema, quedando así implementado el módulo de acceso y de recuperación de casos.
4. El método de adaptación propuesto puede ser visto como un complemento para la adaptación nula, pues además de apoyar las decisiones del usuario, presentándole experiencias previas similares al nuevo problema para que éste conforme la solución final, muestra al usuario una descripción conceptual del conjunto que conforman los casos recuperados, facilitando un análisis objetivo para la toma de decisiones.
5. El procedimiento para determinar la solución descrito en el algoritmo 6, es un método para la adaptación transformacional por sustitución, donde la estructura de la solución permanece inalterable, sólo cambian los valores de los atributos por lo que implica recalcular algunos parámetros de la solución encontrada, los cuales son sustituidos por los conceptos con mayor similitud al caso a resolver.

Adicionalmente, se puede concluir que el modelo proporciona una vía de evaluación de la solución dada en el proceso de adaptación, que facilita disponer de diferentes soluciones al mismo problema y compararlos. Por último, el modelo puede ser útil en la reincorporación del nuevo problema a la base de casos a través del proceso

de actualización de la memoria de casos, para su posterior reutilización, paso más importante del razonamiento basado en casos, ya que incorpora inteligencia a lo que, de un modo contrario, sería un mero proceso de reconocimiento de patrones.

Referencias

- [1] I. O. Ayaquica-Martínez, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, "Conceptual K-Means Algorithm with Similarity Functions," in *Progress in Pattern Recognition, Image Analysis and Applications: 10th Iberoamerican Congress on Pattern Recognition, CIARP 2005, Havana, Cuba, November 15-18, 2005. Proceedings*, A. Sanfeliu and M. L. Cortés, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 368–376.
- [2] N. M. Sánchez, M. M. G. Lorenzo, and Z. Z. G. Valdivia, "Modelo para diseñar sistemas de enseñanza-aprendizaje inteligentes utilizando el razonamiento basado en casos," *Avances en Sistemas e Informática*, vol. 6, no. 3, pp. 67–78, 2009.
- [3] Y. Rodríguez and M. García, "Generalización de la métrica basada en la diferencia de valores (VDM) para variables lingüísticas y su aplicación en sistemas basados en el conocimiento," Tesis presentada en opción del grado científico de Doctor en Ciencias Técnicas. UCLV, 2007.
- [4] I. G. Martínez, R. E. B. Pérez, and A. T. Rodríguez, *Un sistema basado en casos para la toma de decisiones en condiciones de intertidumbre*. Editorial Universitaria, 2002.
- [5] B. Pérez and E. Rafael, *Sistemas Basados en el Conocimiento*. 1998.
- [6] J. Ruiz-Shulcloper, "Reconocimiento lógico combinatorio de patrones: teoría y aplicaciones," 2009.
- [7] X. E. Olvera-Rocha and M. R. Ortiz-Posadas, "Diagnóstico Diferencial de Glaucoma Mediante el Enfoque Lógico-Combinatorio de Reconocimiento de Patrones," in *V Latin American Congress on Biomedical Engineering CLAIB 2011 May 16-21, 2011, Habana, Cuba: Sustainable Technologies for the Health of All*, J. Folgueras Méndez, Y. T. Aznielle Rodríguez, F. C. Calderón Marín, B. S. Llanusa Ruiz, J. Castro Medina, H. Vega Vázquez, M. Carballo Barreda, and R. Rodríguez Rojas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 674–677.
- [8] M. García-Borroto, "Searching extended emerging patterns for supervised classification," in *Proceedings of the 10th international conference on Intelligent data engineering and automated learning*, 2010.
- [9] R. S. Michalski and R. B. Stepp, "Concept-based clustering versus numerical taxonomy," 1981.
- [10] R. S. Michalski, "Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts," *Journal of Policy Analysis and Information Systems*, vol. 4, no. 3, pp. 219–244, 1980.
- [11] R. S. Michalski and R. E. Stepp, "Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 4, pp. 396–410, Jul. 1983.
- [12] R. E. Stepp and R. S. Michalski, "Conceptual clustering: Inventing goal-oriented classifications of structured objects," *Machine learning: An artificial intelligence approach*, vol. 2, pp. 471–498, 1986.
- [13] S. J. Hanson and M. Bauer, "Conceptual clustering, categorization, and polymorphy," *Machine Learning*, vol. 3, no. 4, pp. 343–372, 1989.
- [14] H. Ralambondrainy, "A conceptual version of the K-means algorithm," *Pattern Recognition Letters*, vol. 16, no. 11, pp. 1147–1157, Nov. 1995.
- [15] J. F. Martínez-Trinidad and G. Sánchez-Díaz, "LC: A Conceptual Clustering Algorithm," in *Machine Learning and Data Mining in Pattern Recognition: Second International Workshop, MLDM 2001 Leipzig, Germany, July 25–27, 2001 Proceedings*, P. Perner, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 117–127.
- [16] A. Pons-Porrata, J. Ruiz-Shulcloper, and J. . Martínez-Trinidad, "RGC: A new conceptual clustering algorithm for mixed incomplete data sets," *Mathematical and Computer Modelling*, vol. 36, no. 11–13, pp. 1375–1385, Diciembre 2002.
- [17] W. D. Seeman and R. S. Michalski, "The CLUSTER3 system for goal-oriented conceptual clustering: method and preliminary results," 2006, vol. 1, pp. 81–90.
- [18] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, vol. 2, no. 2, pp. 139–172, 1987.
- [19] M. Lebowitz, "Experiments with incremental concept formation: UNIMEM," *Machine Learning*, vol. 2, no. 2, pp. 103–138, 1987.
- [20] E. A. Feigenbaum, "The simulation of verbal learning behavior," in *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, 1961, pp. 121–132.

- [21] J. H. Gennari, P. Langley, and D. Fisher, "Models of incremental concept formation," *Artificial Intelligence*, vol. 40, no. 1, pp. 11–61, Sep. 1989.
- [22] J. Béjar and U. Cortés, "LINNEO+: Herramienta para la adquisición de conocimiento y generación de reglas de clasificación en dominios poco estructurados," in *Actas del III Congreso Iberoamericano de Inteligencia Artificial (IBERAMIA92)*, 1992.
- [23] C. Carpineto and G. Romano, "GALOIS: An order-theoretic approach to conceptual clustering," in *Machine Learning Proceedings 1993*, San Francisco (CA): Morgan Kaufmann, 1993, pp. 33–40.
- [24] K. McKusick and K. Thompson, "Cobweb/3: A portable implementation," 1990.
- [25] J. F. Martínez-Trinidad and J. Ruiz-Shulcloper, "LC-Conceptual Algorithm: Characterization using typical testors by class," in *Proceedings of the 7th European Congress on Intelligent Techniques & Soft Computing*, 1999.
- [26] J. Ruiz-Shulcloper, A. Guzmán Arenas, and J. F. Martínez-Trinidad, *ENFOQUE LÓGICO COMBINATORIO AL RECONOCIMIENTO DE PATRONES*. México, 1999.
- [27] Ramón C. Pico Peña, "Determinación del umbral de semejanza Bo para los algoritmos de agrupamiento lógico-combinatorios mediante el dendrograma de un algoritmo jerárquico," presented at the SIARP'99, Ciudad de La Habana, 1999.
- [28] R. Lopez De Mantaras, D. Mcsherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T. Cox, K. Forbus, M. Keane, A. Aamodt, and I. Watson, "Retrieval, reuse, revision and retention in case-based reasoning," *The Knowledge Engineering Review*, vol. 20, no. 03, pp. 215–240, Sep. 2005.
- [29] Y. S. Alganza and A. P. Porrata, "LEX: Un nuevo algoritmo para el cálculo de los Testores Típicos," *Ciencias Matemáticas*, vol. 21, no. 1, 2003.
- [30] E. Ramentol-Martínez, F. Herrera, R. B. Pérez, Y. C. Mota, and Y. S. López, "Edición de Conjuntos de Entrenamiento no Balanceados, haciendo uso de Operadores Genéticos y la Teoría de los Conjuntos Aproximados," presented at the VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, 2009.