UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2

ALGORITMO CONCEPTUAL PARA LA CLASIFICACIÓN SUPERVISADA EN EL RECONOCIMIENTO LÓGICO COMBINATORIO DE PATRONES

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Yoandy Antonio Figuerola Díaz

Tutoras: MSc. Yunia Reyes González
Dra.C. Natalia Martínez Sánchez

La Habana, 2017

[…] nada satisface más al hombre que los éxitos que se logran con su esfuerzo y con su trabajo […]

Fidel Castro, 1965

Dedicatoria

Le dedico esta tesis de investigación a: Digna Figueroa Ferrer y a mis padres Elena y Manuel, por darme el apoyo y las fuerzas necesarias para continuar en mis estudios académicos.

Agradecimientos

Le agradezco en primera instancia a Di-os por darme la posibilidad de vivir en un país libre y soberano, a mi abuela querida Digna Figueroa, ya que sin sus consejos y enseñanzas no hubiese alcanzado las metas que hasta ahora me he propuesto, a mi padres Elena Díaz y Manuel Antonio quienes me han dado su apoyo incondicional y han estado presente en cualquier circunstancia de mi vida, a mis tutoras Yunia Reyes Gonzales y Natalia Martinez Sanchez quienes me encaminaron en el área del saber y aportaron mucho a mi vida como profesional, su precaución, paciencia y tiempo que dedicaron para guiarme, mostraron que la sabiduría no tiene fronteras, de las cuales me siento muy agradecido, a la rectora Miriam Nicado quien siempre se ha preocupado por todos los estudiantes y trabajadores de la universidad, a mis hermanos, sobrinos, primos, compañeros vecinos, a mis amigos: Guillermo Dagnesse, Luis Ferrer, Katerine, Luis Manuel, Eral, Damian Troya, Denis Eugene, Castrillon, Alexander Delgado, Adrian Gainza, quienes siempre han estado a mi lado (en las buenas y malas) apoyándome en cualquier decisión, a los profesores: Madelin Haro, Abel Velázquez, Frank Ruben, Lester Vallejo, Lester González, Yanelis Benitez, Mirta Beltrandez, Daniel Sampedro, Maidelis Milanés, Ernesto Yero, Hector González, quienes me mostraron con su modo de actuar, que un mejor camino es posible.

Muchas Gracias.

DECLARACIÓN DE AUTORÍA

Declaro que yo: **Yoandy Antonio Figuerola Díaz** soy el único autor de este trabajo: **Algoritmo conceptual para la clasificación supervisada en el reconocimiento lógico combinatorio de patrones,** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de junio del año 2017.

_____

Yoandy Antonio Figuerola Díaz

_____ _____

Dra. C. Natalia Martínez Sánchez *MSc. Yunia Reyes González*

RESUMEN

La investigación que se realiza se fundamenta teóricamente en el reconocimiento lógico combinatorio de patrones dando continuidad al resultado de investigaciones científicas del grupo de Inteligencia Artificial y Reconocimiento de Patrones de la Universidad de las Ciencias Informáticas, la cual tiene dentro de sus misiones desarrollar aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

En este trabajo se expone un nuevo algoritmo de clasificación supervisada basado en los conceptos generados utilizando algoritmos de agrupamiento conceptual, el cual es una alternativa a tener en cuenta en problemas de apoyo a la toma de decisiones en cualquier área del saber.

El algoritmo de clasificación conceptual que se presenta se implementa en la herramienta conceptual HCC, la cual ofrece otras funcionalidades que permiten trabajar con problemas no supervisados y realizar acciones comparativas de los resultados con dos algoritmos de clasificación supervisada clásicos del reconocimiento lógico combinatorio.

Finalmente, se presenta el esquema de estudio experimental que se aplica para validar el algoritmo ACC y un caso de estudio relacionado con el proceso de solicitud y selección de las asignaturas del currículo optativo del Plan de Estudios de la carrera de Ingeniería en Ciencias Informáticas.

Palabras clave: agrupamiento conceptual, reconocimiento lógico combinatorio de patrones.

Abstract

The research that is carried out is based theoretically on the logical combinatorial recognition of patterns that gives continuity to the result of scientific investigations of the group of Artificial Intelligence and Pattern Recognition of the University of the Computer Sciences, which has in its missions applications and services of the Development, based on the study-work linkage as a training model.

This work presents a new supervised classification algorithm based on concepts generated using conceptual clustering algorithms, which is an alternative to take into account in problems of support to decision making in any area of knowledge.

The conceptual classification algorithm that is presented is implemented in the conceptual tool HCC, which offers other functionalities that allow to work with unsupervised problems and to perform comparative actions of the results with two classical supervised classification algorithms of combinatorial logic recognition.

Finally, we present the experimental study scheme that is applied to validate the ACC algorithm and a case study related to the application and selection process of the elective curriculum subjects of the Curriculum of the Engineering in Computer Science.

Key words: conceptual grouping, logical combinatorial pattern recognition.

TABLA DE CONTENIDOS

INTRODUCCIÓN

A partir de la década de los noventas, se observa un explosivo crecimiento de las capacidades para emplear, generar y colectar datos. Sin embargo, la accesibilidad a grandes volúmenes de datos, que en muchas ocasiones es un gran problema, no es el único ni el más complejo. Lo que se convierte en un gran problema es el poder procesarlos, interpretarlos y sobre la base de esas interpretaciones poder actuar en aras de los intereses en cuestión. Con este propósito se comienza a desarrollar en esos años una nueva área de investigaciones: la Minería de Datos, que se puede identificar como la parte del proceso de Descubrimiento de Conocimientos a partir de Datos. En el proceso de Descubrimiento del Conocimiento a partir de Datos (KDD, por su sigla en inglés), una de las tareas más importantes en cuanto a su repercusión, es la clasificación de los datos. Esto hace que esta disciplina tenga un nexo ineludible con el reconocimiento de patrones. Es decir, el procesamiento de datos es un factor común entre estas dos disciplinas.

Por problemas de reconocimiento de patrones, en este trabajo, se entienden todos aquellos relacionados con la clasificación de objetos y fenómenos y con la determinación de los factores que inciden en los mismos. Así, se consideran cuatro familias de problemas que se denominan respectivamente (Ruiz-Shulcloper 2009):

a. selección de rasgos;
b. clasificación supervisada;
c. clasificación no supervisada;
d. clasificación parcialmente supervisada.

La clasificación supervisada es el problema más conocido del reconocimiento de patrones (Ruiz-Shulcloper 2009) y con el que más se está relacionado pues mucha de la actividad humana está vinculada, directa o indirectamente, con procesos de clasificación. Dado un universo de objetos y el conocimiento acerca de la existencia de ciertas clases con características (propiedades) de especial interés y una muestra de objetos que pertenecen a cada una de ellas, el problema consiste en determinar para cada uno de los objetos no clasificados las relaciones de pertenencia de los mismos con cada una de las clases.

En muchas disciplinas como la Medicina, Sociología, Geociencias, Criminalística, frecuentemente se presentan problemas de clasificación, de diagnóstico, de pronóstico, de determinación de factores de influencias. El reconocimiento lógico combinatorio de patrones (RLCP) proporciona un marco teórico para el estudio y solución de este tipo de problemas, y su esencia consiste en trabajar con datos numéricos y no numéricos de manera simultánea en los objetos, considerando además la ausencia de información.

El reconocimiento de patrones es una ciencia con un fuerte carácter aplicado e interdisciplinario. Está relacionado con procesos (ingenieriles, físicos, matemáticos y computacionales) de datos (entendidos en la concepción general antes esbozada) que provienen de objetos físicos (fotos, hologramas, escrituras, jeroglíficos, símbolos, señales bioeléctricas, acústicas, etc.) y/u objetos abstractos (nuplos de un producto cartesiano de conjuntos – duros, difusos, rugosos) con el propósito de obtener (por medio de dispositivos computacionales y/o seres humanos) información

que permita establecer las propiedades de ciertos subconjuntos de objetos y/o las relaciones entre ellos (Ruiz-Shulcloper 2009).

Existen diferentes enfoques para resolver el problema de la clasificación supervisada; Aprendizaje basado en instancias, Redes Neuronales Artificiales, Algoritmos genéticos, Redes Bayesianas, Reconocimiento Estadístico, Reconocimiento lógico combinatorio de patrones, entre otros. Así como, disímiles aplicaciones donde han sido utilizados estos enfoques con resultados satisfactorios, cuya causa fundamental del éxito está dada por la pertinencia en la selección del paradigma a utilizar dada las características del problema a resolver (Rodríguez Sarabia, García Lorenzo, y De Baets, 2007), (Martínez Sánchez, García Valdivia, y García Lorenzo, 2009).

En este trabajo se selecciona el reconocimiento lógico combinatorio de patrones dando continuidad al resultado de investigaciones científicas del grupo de Inteligencia Artificial y Reconocimiento de Patrones de la Universidad de las Ciencias Informáticas (UCI) (Reyes González y Martínez Sánchez 2014), (González, Martínez, y Díaz, 2014), (Reyes González y Martínez Sánchez, 2016) la cual tiene dentro de sus misiones desarrollar aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

La UCI cuenta con una red de centros de desarrollo de software agrupados en cinco grandes áreas del saber; Salud, Educación, Administración Pública, Telemática y Empresa-Industria que se comercializan bajo la estrategia marcaria XAVIA, XAUCE, XEDRO, XILEMA Y XABAL respectivamente. La introducción de los resultados

científicos de esta investigación en los softwares que se desarrollan en los centros antes mencionados permite dar un valor agregado a los mismos influyendo tanto en su eficacia como en su nivel de comercialización.

Los problemas de clasificación con aprendizaje o clasificación supervisada se han enfrentado desde varios modelos matemáticos, entre ellos; los algoritmos tipo Kora, Modelo de Algoritmo de Votación (MAV), conjunto de representantes (CR), Algoritmo basado en el Peso de los Patrones (APP), Algoritmo basado en el Ideal de la Clase (AIC), Algoritmo basado en Umbral de Exactitud (AUE) (Ruiz-Shulcloper 2009).

Por otro lado, están los problemas clasificación sin aprendizaje (clasificación no supervisada), en los que no se conoce cómo se agrupan los objetos. Para la solución de este tipo de problema existen, entre otros, los modelos de algoritmos conceptuales como el LC-Conceptual, (Martínez-Trinidad y Ruiz-Shulcloper, 1998), RGC (Pons-Porrata, Ruiz-Shulcloper y Martínez-Trinidad, 2002), (Martnez-Trinidad y Sánchez-Díaz, 2001), K-means Conceptual (Ayaquica-Martínez, Martínez-Trinidad y Carrasco-Ochoa 2005).

Otro problema importante en el reconocimiento de patrones, es el de clasificación con aprendizaje parcial (clasificación parcialmente supervisada), que es análogo al de clasificación con aprendizaje, excepto que hay una clase de objetos de la que no se tiene una muestra.

Hasta el momento los problemas de clasificación supervisada en el marco del RLCP se han resuelto utilizando el Algoritmo basado en el Ideal de la Clase

(AIC), en Umbrales de Exactitud (AUE), en la tipicidad y el contraste, en conjunto de representantes y por último se desarrolló el modelo KORA (Ruiz-Shulcloper 2009).

Los anteriores algoritmos presentan limitaciones como por ejemplo en el caso del algoritmo Ideal de la clase, para clasificar el objeto en la clase se basa precisamente en el objeto de mayor peso informacional y no siempre este es el objeto más representativo de la clase. La dependencia de los umbrales para poder realizar la clasificación es una de las limitantes del algoritmo AUE. Por su parte los algoritmos basados en la tipicidad y el contraste trabajan sólo con variables reales y no admite ausencia de información, asumiendo que todos los rasgos aportan la misma información. No permite un tratamiento diferenciado de los rasgos al admitir sólo un criterio de comparación de los valores de los mismos. La semejanza entre los objetos se restringe a una sola función. El algoritmo se limita a determinar cuatro tipos de objetos en cada una de las clases, es decir, no pasa de ser un algoritmo para el análisis de los datos y no un algoritmo de clasificación supervisada (Ruiz-Shulcloper 2009).

Para superar estas limitaciones se desarrolla el algoritmo Kora-Ω, basado en el concepto de rasgos complejos, el cual admite cualquier tipo de variable, no sólo las booleanas. Estas variables pudieran ser descripciones heterogéneas, es decir, con datos mezclados e incompletos y la ausencia de información es admisible. Esto implica que los criterios de comparación de valores de cada una de los rasgos ni tienen que ser los mismos, ni tienen que ser la igualdad estricta. Cualquier tipo de función de similitud puede ser usada para comparar las sub-

descripciones de los objetos. No obstante, no incluye ningún algoritmo para encontrar los rasgos complejos, y éstos son buscados exhaustivamente. La búsqueda se realiza en subconjuntos de atributos previamente seleccionados, pero no se propone una manera de encontrarlos automáticamente, siendo éstos definidos por el usuario. Estas limitaciones afectan la eficacia de la clasificación en los problemas supervisados.

Debido a lo expuesto anteriormente se plantea el siguiente **problema a resolver**: ¿Cómo mejorar la eficacia en la clasificación de objetos para problemas supervisados en el marco del Reconocimiento Lógico Combinatorio de Patrones?

El problema se enmarca en el **objeto de estudio:** La clasificación supervisada en el enfoque lógico-combinatorio de patrones, siendo así el **campo de acción**: Algoritmos conceptuales para la clasificación de objetos.

Para dar solución al problema planteado, se propone como **objetivo general:** Desarrollar un algoritmo para problemas de clasificación supervisada en el reconocimiento lógico combinatorio de patrones, basado en la teoría del agrupamiento conceptual.

Para cumplir con los objetivos propuestos se plantean los siguientes **objetivos específicos:**

a. Elaborar el marco teórico referencial de la investigación, relacionado con los algoritmos para la clasificación supervisada en el RLCP.

b. Formalizar el algoritmo propuesto de clasificación supervisada utilizando los conceptos generados en la etapa intencional de los algoritmos de agrupamiento conceptual del RLCP.
c. Implementar computacionalmente el algoritmo de clasificación supervisada propuesto en una herramienta de autor.
d. Evaluar el algoritmo propuesto en la solución de problemas prácticos.

Para darle cumplimiento a los objetivos específicos se plantean las siguientes **Tareas de Investigación:**

a. Caracterización de los algoritmos de clasificación supervisada del enfoque lógico-combinatorio del reconocimiento de patrones.
b. Caracterización de los algoritmos de agrupamiento conceptuales.
c. Caracterización de las metodologías, herramientas, tecnologías y lenguaje de programación para guiar el desarrollo de software.
d. Implementación del algoritmo conceptual de tipo supervisado basado en los conceptos
e. Análisis de la eficiencia del algoritmo propuesto.

Para el desarrollo de las tareas científicas se han combinado diferentes **métodos y procedimientos teóricos y empíricos** de la investigación científica en la búsqueda y procesamiento de la información. Los fundamentales son:

Métodos Teóricos:

a. **Analítico–Sintético:** Este método es utilizado para realizar el análisis de documentos y teorías, permitiendo la extracción de los elementos más

importantes que se relacionan con los algoritmos conceptuales para la clasificación supervisada. También se emplea este método para profundizar en las tecnologías, herramientas y metodologías a utilizar en el desarrollo de la aplicación.

b. **Inductivo-Deductivo:** Este método permite llegar al planteamiento del objetivo, además de la extracción de las ideas fundamentales para la elaboración y fundamentación teórica del trabajo de diploma.

c. **Modelación:** Este método es utilizado para representar gráficamente los diferentes elementos que componen el diseño del algoritmo y confeccionar los modelos y diagramas asociados al desarrollo del mismo.

d. **Histórico-lógico:** Este método es utilizado para estudiar el comportamiento de los algoritmos, específicamente de LC-Conceptual, RGC y Kora-Ω a través de los años y analizar su funcionamiento con el fin de dar cumplimiento al objetivo de la investigación.

Métodos Empíricos:

a. **Entrevista:** Se realizaron entrevistas a los expertos para determinar los aspectos importantes del algoritmo conceptual a desarrollar y los criterios a tener en cuenta para su valoración.

Después de la revisión de la literatura y el desarrollo consecuente del marco teórico se formula la siguiente idea a defender: El desarrollo de un algoritmo para problemas de clasificación supervisada en el reconocimiento

lógico combinatorio de patrones, basado en la teoría del agrupamiento conceptual, contribuye a mejorar la eficacia en la clasificación de nuevos objetos.

La memoria escrita se estructura en tres capítulos. En el primero se describen los referentes teóricos relacionados con los conceptos básicos del RLC con énfasis en los algoritmos de clasificación supervisada existentes y en los algoritmos de agrupamiento conceptual. Además, se describen tecnologías, Framework, metodología y lenguajes de programación que se adecuan a las características de la herramienta computacional que se implementa como resultado práctico de esta investigación.

En el segundo capítulo se describe el algoritmo de clasificación supervisada utilizando los conceptos del agrupamiento conceptual y la validación del de éste a través de la comparación con dos algoritmos de clasificación supervisada del RLCP. En el tercer y último capítulo se describe la herramienta computacional HCC y describe una aplicación utilizando

Por último, se muestran las conclusiones del trabajo, las recomendaciones que sirven como base para continuar perfeccionando los resultados que se muestran y la bibliografía científica utilizada.

CAPÍTULO 1. ACERCA DE LA CLASIFICACIÓN SUPERVISADA EN EL RLCP Y HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE

En este capítulo se describen los conceptos básicos del RLCP fundamentalmente dirigido a uno de los cuatros problemas fundamentales a los que se dedica este paradigma, la clasificación supervisada. Se describen las herramientas computacionales necesarias para la implementación del software que se desarrolla. Por último, se describen las consideraciones parciales a las que arriba el autor después de consultar la literatura científica referenciada.

1.1 Reconocimiento lógico combinatorio de patrones

En algunas disciplinas tales como: (Medicina, Sociología, Geociencias, Criminalística) frecuentemente se presentan problemas de clasificación, de diagnóstico, de pronóstico, de determinación de factores de influencias. El Reconocimiento Lógico Combinatorio de Patrones (RLCP) (Ruiz-Shulcloper 2009) proporciona un marco teórico para el estudio y solución de este tipo de problemas, y su esencia consiste en trabajar con datos numéricos y no numéricos de manera simultánea en los objetos, considerando además la ausencia de información.

Por problemas de reconocimiento de patrones se entienden todos aquellos relacionados con la clasificación de objetos y fenómenos y con la determinación de los factores que inciden en los mismos (Pons-Porrata, Ruiz-Shulcloper y Martínez-Trinidad, 2002). Por otro lado, en muchas ciencias aplicadas está presente el problema de revelar la estructura subyacente en una colección de objetos. Como

una alternativa para la solución de este problema surge a partir de los trabajos de R. S. Michalski en la década de los 80 el agrupamiento conceptual. Este enfoque propone no sólo determinar la estructuración para un conjunto de objetos, sino además aportar la propiedad o concepto que caracteriza a los grupos formados en la estructuración.

El primer modelo de algoritmos conceptuales fue propuesto por Michalski a inicio de la década de los ’80 (Michalski 1980), dando origen a la familia de algoritmos CLUSTER, que fueron los pioneros en la solución del agrupamiento conceptual. Posteriormente se desarrollaron otros permitiendo clasificarlos en dos grupos: los incrementales y los no incrementales (Guerra-Gandón, Vega-Pons, y Ruiz-Shulcloper, 2012).

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 (Ruiz-Shulcloper 2009): UNIMEM , COBWEB y Galois, LINNEO+ , LABYRINTH.

Los algoritmos no incrementales estructuran una muestra de objetos utilizando el conjunto de datos completo. Ejemplos de estos algoritmos son: CLUSTER/2, y

CLUSTER/S, WITT, K-Means conceptual, LC-Conceptual (Martínez-Trinidad 2001) y RGC (Pons-Porrata, Ruiz-Shulcloper y Martínez-Trinidad, 2002).

En (Pons-Porrata, Ruiz-Shulcloper y Martínez-Trinidad, 2002) se realiza un análisis crítico de los principales trabajos publicados en el área del agrupamiento conceptual de objetos. Se presenta una comparación cualitativa de acuerdo a un conjunto de características deseables en este tipo de algoritmos de agrupamiento y se señala que la mayoría de los algoritmos conceptuales no son aplicables en problemas reales en los cuales, por lo general, los objetos se describen por rasgos mezclados e incompletos. Del anterior análisis, resulta claro que los algoritmos LC-Conceptual y RGC resuelven esta problemática. Sin embargo, no han sido empleados para abordar problemas de clasificación supervisada donde pudieran tener una interesante aplicación. En tal sentido, este trabajo propone un modelo de algoritmos basado en la idea esencial de la formación de los conceptos propuestas por LC-Conceptual para la solución de problemas de clasificación en el RLCP.

1.2 Clasificación supervisada en el RLCP

Dado un universo de objetos estructurados en clases, de cada una de las cuales se tiene una muestra no vacía, un problema de **clasificación supervisada** consiste en, construir un algoritmo que permita, a partir de esta muestra, decidir a cuál de las clases pertenece un objeto nuevo que se quiera clasificar.

Los problemas de clasificación supervisada dentro del Reconocimiento Lógico Combinatorio de Patrones se describen formalmente de la siguiente manera (Ruiz-Shulcloper 2009):

Definición 1.1: Dada una información estándar verdadera I0(K1,...,Kr) de las clases K1′ ,...,Kr′ donde Ki⊂K'i , i=1,...,r, que también se denomina muestra ó matriz de entrenamiento; dada la sucesión I(O′1),...,I(O′q) de descripciones estándar (que pudieran no ser completas) de objetos admisibles O′1,..., O′q (se llama muestra o matriz de control). El problema de la clasificación supervisada consise en hallar un algoritmo A tal que: A(I0(K1,...,Kr),I(O1′),...,I(Oq′))= || αAj (Oi′)||q× r donde α jA (O′i)∈{0, 1,∗} es la respuesta del algoritmo A en cuanto a la pertenencia de O′i a las clases Kj,||...|| q×r denota una matriz de q filas y r columnas, ∗ denota la abstención del algoritmo A al clasificar a O.

En particular un algoritmo de clasificación supervisada es un procedimiento efectivo que cumple las condiciones siguientes:

a. Requiere de una información estándar I (K_1,...,K_r) correcta para las clases K_1′,…,K_r′ de un universo U dado.

b. Requiere de un criterio de comparación entre descripciones de los objetos de U.

c. Admite como entrada, la descripción de un objeto admisible de U en términos de los mismos rasgos que la información estándar.

d. Devuelve como salida un r-uplo de pertenencia.

Por clasificación de un objeto O∈U por un clasificador supervisado A, se entiende la acción de asignar un r-uplo de pertenencias a las clases de U, tomando dicho objeto como entrada de A. En algunos textos también se entiende por clasificación el resultado de esta acción. La clase que A, con matriz de entrenamiento M, le asigna a un objeto O se denotará por αA(M,O). A esto último se le llamará r-uplo de pertenencia del objeto asignado por A. En la descripción de los algoritmos se utiliza la notación funcional de un clasificador supervisado (Ruiz-Shulcloper 2009).

Dentro del RLCP se han desarrollado diversos algoritmos de clasificación supervisada como (Ruiz-Shulcloper 2009): ALVOT, KORA-{Ω} y CR+. En este trabajo se exponen las principales ideas de los más representativos.

1.2.1 Algoritmo basado en el ideal de la clase (AIC)

El algoritmo AIC calcula el peso informacional de los objetos de cada clase y selecciona como ideal, aquel objeto que alcance el mayor peso informacional en su clase. Luego calcula la semejanza entre el nuevo objeto a clasificar y los ideales de cada clase, mediante una función de semejanza. Por último, el nuevo objeto se clasifica en la clase a la cual pertenece el objeto ideal, que obtuvo la mayor semejanza con él. Si sucede que los valores de semejanza son iguales para todas las clases el algoritmo puede abstenerse o multiclasificar, dependiendo de la aplicación. Una limitación de este algoritmo es que para clasificar el objeto en la clase se basa precisamente en el objeto de mayor peso informacional y no siempre ese es el objeto más representativo de la clase.

1.2.2 Algoritmo basado en umbral de exactitud (AUE)

El algoritmo AUE de clasificación basado en umbral de exactitud calcula el peso informacional de todos los objetos de la matriz de entrenamiento y del nuevo objeto con respecto a cada una de las clases. Luego se calculan umbrales para cada clase garantizando que los pesos informacionales de la mayoría de los objetos de esa clase estén comprendidos entre esos dos umbrales, de la forma: si $\varepsilon_{v-1} < \psi_v(Oj) < \varepsilon_v$ entonces $O_j \in K_v$. Dado un nuevo objeto $O \in U$, si $\varepsilon_{t-1} < \psi_t(O) < \varepsilon_t$ entonces $O \in K_t$. Si se presenta el caso en que $\psi_t(O) = \varepsilon_t$; t=1, ..., r; el algoritmo puede abstenerse o multiclasificar, dependiendo de la aplicación. Este algoritmo presenta como limitación la dependencia de los umbrales para poder realizar la clasificación.

1.2.3 Algoritmo basado en la tipicidad y el contraste

La tipicidad de un objeto en su clase, viene expresada por la frecuencia en dicha clase, de los valores de los rasgos que aparecen en su descripción. Mayor frecuencia, mayor tipicidad. El contraste da una medida de cuán diferente es la tipicidad de un objeto de una clase determinada con respecto a la tipicidad del mismo objeto correspondiente a cada una de las restantes clases. Estos objetos se pueden clasificar en propios, generales, atípicos y no propios.

- Propios: Son aquellos que se asemejan suficientemente a los objetos de su clase (característicos) y se diferencian significativamente de los objetos de las restantes clases.
- Generales: Son objetos característicos de la mayoría de las clases.
- Atípicos: No son característicos de clase alguna.

- No Propios: Son objetos no característicos de la clase donde están ubicados.

Este algoritmo calcula los pesos informacionales de los rasgos a partir de los criterios de comparación de los rasgos y la función de semejanza definidos para el problema. Luego determina la cantidad de tipos de objetos por clases con los que se trabaja, así como las reglas sobre la base de las cuales se decide el tipo de cada objeto de M. En el siguiente paso se calcula el peso informacional $\psi_V(O)$ y el diferenciante $\Psi_V^D(O)$ de los m objetos y a continuación la tipicidad de cada objeto de M para cada una de las clases $K_1,\ldots,K_r$, por medio de la expresión (1), donde $T_{j,i}(O_h)$, es el promedio de los pesos informacionales de los objetos que estando en K_j, son semejantes a O_h y $T_{j,0}(O_h)$: es el promedio de los pesos informacionales de los objetos que estando en K_j, no son semejantes a O_h

$$T_j(O_h) = \frac{T_{j,1}(O_h) + \sum_{\substack{i=1 \\ i \neq j}}^{r} T_{i,0}(O_h)}{T_{j,0}(O_h) + \sum_{\substack{i=1 \\ i \neq j}}^{r} T_{j,1}(O_h)} \qquad (1)$$

A continuación se calcula la tipicidad de cada objeto de M para cada una de las clases $K_1, \ldots, K_r$, por medio de las expresiones (2) y (3), donde D es el conjunto de valores de semejanza en los que se considera que los objetos son similares y $\Psi_j^D(O_i)$ es el peso diferenciante de O_j

$$T_{j,1}(O_h) = \frac{1}{|K_j|} \sum_{\substack{O_i \in K_j \\ \Gamma(O_h, O_i) \in D}} \Psi_j(O_i) \qquad (2)$$

$$T_{j,0}(O_h) = \frac{1}{|K_j|} \sum_{\substack{O_i \in K_j \\ \Gamma(O_h, O_i) \notin D}} \Psi_j^D(O_i) \qquad (3)$$

El contraste de cada objeto de M para cada una de las clases K_1, …, K_r, es calculado por medio de la expresión (4), siendo p* la clase donde O_h alcanza el máximo de tipicidad y $T(O_h)=\max\{T_1(O_h),\ldots, T_r(O_h)\}$

$$C(O_h) = \sqrt{\frac{\sum_{\substack{j=1 \\ j \neq p^*}}^{r} \left(T(O_h) - T_j(O_h)\right)}{r-1}} \qquad (4)$$

Finalmente, el algoritmo puede concluir si el objetivo era hacer un análisis de la estructura interna de las clases de M en términos de las relaciones entre los objetos atendiendo a su relevancia informacional determinada por las expresiones correspondientes de los pesos antes descritos. Si, por el contrario, el propósito es la clasificación de nuevos objetos de U\M, entonces se determina la tipicidad del objeto a clasificar con respecto a cada una de las *r* clases, así como su contraste. Los objetos serán clasificados en aquellas clases en las que se hayan obtenido la máxima tipicidad.

El algoritmo se limita a determinar los cuatro tipos de objetos en cada una de las clases, es decir, no pasa de ser un algoritmo para el análisis de los datos en M y no un algoritmo de clasificación supervisada.

1.2.4 Algoritmo basado en Conjuntos de Representantes

La idea de conjuntos de representantes fue introducida en por Baskakova y Zhuravliov. Basándose en el concepto de precedencia parcial, introducen la idea de valorar información a favor y en contra de la pertenencia de los objetos a las clases, así como la de considerar que los parámetros utilizados para la clasificación deben estar asociados a cada clase.

El algoritmo CR, asume que todos los rasgos son booleanos. Sea un sistema de conjuntos de apoyo para la clase K_j, $\Omega \in \Omega K_j$. Por conjunto de representantes positivos para la clase K_j con respecto a Ω, se entenderá el conjunto de todos los valores, para la Ω - parte correspondiente, que aparecen η_j veces en las Ω - partes de los objetos de K_j y no aparecen ni una vez en las Ω - partes de los objetos de CK_j. Cada elemento de este conjunto será llamado un representante positivo.

Por conjunto de representantes negativos para la clase K_j con respecto a Ω, se entenderá el conjunto de todos los valores, para la Ω - parte correspondiente, que aparecen η_j veces en las Ω - partes de los objetos de CK_j y no aparecen ni una vez en las Ω - partes de los objetos de K_j. Cada elemento de este conjunto será llamado un representante negativo.

Por conjunto de representantes neutrales para la clase K_j con respecto a Ω, se entenderá el conjunto de todos los valores, para la Ω - parte correspondiente, que aparecen η_j veces en las Ω - partes de los objetos de CK_j y no son representantes positivos ni representantes negativos. Cada elemento de este conjunto será llamado un representante neutral.

Este método de clasificación permite establecer una diferenciación entre el valor informacional de los rasgos y los objetos.

Para el funcionamiento del algoritmo se definen los sistemas de conjuntos de apoyo y los parámetros para cada una de las clases. Luego calcula los complementos para cada una de las clases y a continuación los conjuntos de representantes positivos, negativos y neutrales para cada uno de los conjuntos de apoyo de cada una de las clases. Cuando se quiere clasificar un nuevo objeto O se calcula la semejanza, a partir de una función que integra los pesos informacionales de los rasgos y de los objetos y está en dependencia de los conjuntos de representantes positivos, negativos y neutros.

Se calcula el r-uplo informacional del objeto O, el cual tiene la forma ($\alpha_1(O)$, ..., $\alpha_r(O)$), donde:

$$\alpha_j(O) = \begin{cases} 1 \;\; si \;\; \Gamma_j(O) > 0 \\ 0 \;\; si \;\; \Gamma_j(O) < 0 \\ * \;\; si \;\; \Gamma_j(O) = 0 \end{cases}$$

Donde $\alpha_j(O) = *$ significa que el algoritmo se abstiene de clasificar al objeto O.

Existe una extensión del algoritmo para el caso métrico.

1.2.5 Algoritmo Tipo KORA-{Ω}

El modelo de clasificación supervisada KORA-{Ω} se basa en el algoritmo KORA-3 propuesto por M. M. Bongard el cual fue desarrollado para resolver problemas de clasificación en Geofísica y Geología. Este modelo introduce la idea de emplear diferentes niveles de representación. La idea central del modelo es encontrar características complejas que sirvan para identificar objetos de una clase.

Dado R el conjunto de los rasgos en términos de los cuales se definen los objetos de la muestra de entrenamiento en un problema de clasificación supervisada con más de dos clases, no necesariamente disjuntas, el algoritmo KORA-{Ω} basa su

funcionamiento en los conceptos de rasgos complejos, resto y rasgo complejo complementario.

Sea $\Omega = \{x_{i1}, \ldots, x_{ip}\}$ un conjunto de apoyo, y $(a_1,...,a_p)$ una combinación de valores para $x_{i1}, \ldots, x_{ip}$, espectivamente, entonces $(a_1,...,a_p)$ y $x_{i1}, \ldots, x_{ip}$ forman un rasgo β_i-complejo de la clase K_i si y sólo si:

1. $\exists O \in K_i[x_{i1}(O) = a_1 \wedge ... \wedge x_{ip}(O) = a_p]$
2. $\sum_{O \in K_i} \Gamma(\Omega O, (a_1,...,a_p)) \geq \beta_i$
3. $\sum_{O \in CK_i} \Gamma(\Omega O, (a_1,...,a_p)) \geq \lambda_i$, para **i = 1, …, r**

Se denotará por RC (K_i) al conjunto de todos los rasgos β_i-complejos de la clase K_i. Un objeto $O \in K_i$ será denominado δ_i-resto si y sólo si: $\sum_{\Omega rc \in RC(K_i)} \Gamma(\Omega O, \Omega rc) < \delta_1$

Para simplificar la notación, se denota por $r(K_i)$ al conjunto de todos los δ_i-restos de la clase K_i. $(a_1,...,a_p)$ y $x_{i1}, \ldots, x_{ip}$ forman un rasgo β_i-complejo complementario de la clase K_i si y sólo si:

1. $\exists O \in rK_i[x_{i1}(O) = = a_1 \wedge ... \wedge x_{ip}(O) = a_p]$
2. $\sum_{O \in rK_i} \Gamma(\Omega O, (a_1,...,a_p)) \geq \beta_i$
3. $\sum_{O \in CK_i} \Gamma(\Omega O, (a_1,...,a_p)) \geq \lambda_i$, para i = 1,…,r, siendo $\beta_i' < \beta_i$

Se introduce una manera de ponderar los votos provenientes de diferentes rasgos complejos.

Sea r_c un rasgo complejo, para la clase K_i, formado por $(a_1,...,a_p)$ y $x_{i1}, \ldots, x_{ip}$, entonces la ponderación de un rasgo complejo rc, la cual se denotará por $P(r_c)$, estará dada por:

$$P(rc) = \sum_{x_k \in \Omega} \rho(x_k) \sum_{O' \in OC_i(rc)} \Psi_i(O') \qquad (5)$$

donde: $\rho(x_k)$ es el peso informacional del rasgo x_k.

$\Psi_i(O')$ es el peso informacional del objeto O' en la clase K_i.

$OC_i(r_c)$ es el conjunto de los objetos de K_i, caracterizados por r_c.

Con esta ponderación, cuando se quiere clasificar un nuevo objeto O, se puede calcular cuál es la votación que obtiene el objeto O en la clase K_i, la cual se denota por $\Gamma_i(O)$, de la siguiente manera: $\Gamma_i(O) = \sum_{rc \in RC_i(O)} P(rc)$, donde $RC_i(O)$ es el conjunto de los rasgos complejos, de la clase K_i, que caracterizan a O.

Por tanto, el objeto se clasificaría en la clase que obtuviera la votación más alta. Si más de una clase obtiene la votación más alta, el método se abstiene de clasificar al objeto o multiclasifica. La regla de solución que con frecuencia es utilizada es la de mayoría simple, pero el modelo no se restringe a ella, es decir, podría utilizarse cualquier otra regla de decisión. La noción de rasgo complejo está vinculada a otras empleadas en distintas ramas de la ciencia, de manera que resulta como una modelación de dichas nociones. Por ejemplo, los síndromes y los factores de riesgos en la Medicina, son también conjunto de valores de ciertos rasgos que discriminan entre diferentes clases. Estos ayudan a clasificar, pero también a conocer mejor las enfermedades, en el caso de la Medicina.

Basándose en estos preceptos se puede enunciar la definición de regularidad discriminante. Por regularidades se debe entender cualquier combinación de valores de un conjunto de rasgos que describe a un subconjunto de objetos de una muestra dada. Si se trata de una muestra particionada en subconjuntos de objetos, se entenderá por una regularidad discriminante aquella regularidad que describe suficientemente a los objetos de una clase y describe a pocos en cualquier otra clase.

Los rasgos complejos del modelo KORA y los patrones emergentes, son casos particulares de regularidades discriminantes. Los l-complejos, introducidos por Michalski, son también casos particulares de regularidades discriminantes [2].

1.3 Herramientas computacionales existentes

En la actualidad existen múltiples herramientas informáticas que son empleadas para agilizar el proceso de aplicación de la minería de datos. El principal objetivo de estas herramientas es procesar datos y extraer conclusiones sobre patrones de comportamiento. Este proceso es realizado por las herramientas de minería de datos. A continuación, se muestra el estudio realizado a las de código que se asemejan más a esta investigación: Weka, RapidMiner y Keel.

La herramienta Weka[1] es una colección de algoritmos de aprendizaje automático para la solución de problemas de minería de datos del mundo real. Está escrito en Java y se ejecuta en cualquier plataforma. Los algoritmos bien se pueden aplicar directamente a un conjunto de datos o llamadas desde su propio código Java. Es un software que ha sido desarrollado en la universidad de Waikato bajo licencia GPL lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años. Entre las funciones de minería de datos que implementa se encuentran: asociación (Apriori), clasificación (árboles de decisión, vecinos más próximos, Support Vector Machines (SVM)), agrupamiento (K medias, Cobweb) y modelos combinados.

[1] *Herramienta de código abierto escrita en Java. Disponible bajo licencia pública GNU en* http://www.cs.waikato.ac.nz/~ml/weka/

La principal desventaja que posee Weka es que la mayoría de sus funcionalidades sólo son aplicables si todos los datos se mantienen en la memoria principal. Una segunda desventaja es la otra cara de la portabilidad: una implementación de Java puede ser un poco más lento que un equivalente en C / C++ (STROUSTRUP, Bjarne,Addison Wesley 2007). Además, la herramienta Weka sólo posee un algoritmo de agrupamiento conceptual en su clasificación de incremental, es decir, no tiene implementados los algoritmos no incrementales que son los que se estudiarán con mayor énfasis y por donde va estar centrada la solución a esta investigación.

RapidMiner es una de las herramientas más conocida y usada en las empresas. Se ejecuta en todos los sistemas operativos y plataformas más importantes. Se considera la interfaz gráfica de usuario más potente e intuitiva para el diseño de procesos de análisis. Un entorno visual y fácil de usar le permite reconocer errores, aplicar soluciones rápidas, y ver resultados rápidos y precisos, sin necesidad de tamizar a través de código (Burget, R., Karasek, J., Smekal, Z., Uher, V., & Dostal, O. 2010). Además, combina esquemas de aprendizaje y evaluadores de atributos del entorno de aprendizaje Weka. Entre las funciones de minería de datos que implementa se encuentran regresión, clasificación y agrupamiento.

Por otra parte, estas herramientas: Weka y RapidMiner son herramientas libres, pero que tienen la desventaja de tener un proceso engorroso, puesto que requieren tiempo para la preparación y vinculación de los datos con la herramienta, extendiendo así el tiempo de respuesta de los análisis.

Keel es un framework cuyo objetivo es evaluar algoritmos evolutivos con fines educativos y de investigación. Se incluyen regresión, clasificación, *clustering*, minería de patrones. Asimismo, KEEL contiene algoritmos clásicos de extracción, técnicas de pre-procesamiento, y algoritmos de aprendizaje basados en Inteligencia Computacional como aprendizaje basado en reglas evolutivas y modelos híbridos tales como sistemas difusos/genéticos, redes neuronales evolutivas (Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. 2011).

Con el estudio realizado a cada una de estas herramientas y según la bibliografía revisada, se puede concluir que hasta el momento no existe una herramienta que se encargue específicamente del apoyo al modelado del estudiante que implemente algoritmos de agrupamiento conceptual. Por tal motivo, surge la necesidad de desarrollar una herramienta que emplee como técnica algoritmos de agrupamiento conceptual, guiada por el enfoque Lógico-Combinatorio.

1.4 Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y documentos auxiliares que sirven de ayuda a los desarrolladores de software en la realización de nuevas aplicaciones informáticas. En un proyecto de desarrollo, define quién debe hacer qué, cuándo y cómo hacerlo. Es un proceso que puede seguir uno o varios modelos de ciclo de vida e indica cómo hay que obtener los distintos productos parciales y finales en el desarrollo de un software (BOOCH, Grady, et al 1999) (Sommerville 2005).

No existe una metodología de software universal. Las características de cada proyecto exigen que el proceso sea configurable, por lo que actualmente existen

un conjunto de metodologías de desarrollo de software con características específicas.

Según la filosofía de desarrollo, las metodologías se pueden clasificar en:

- **Metodologías tradicionales robustas:** Se basan en la planificación y se aplican a proyectos de gran tamaño respecto a tiempo y recursos. Éstas no ofrecen una buena solución para proyectos donde el entorno es volátil y donde los requisitos no se conocen con exactitud, porque no están pensadas para trabajar con incertidumbre. Son menos orientadas al código y dedican más tiempo pensando en cómo se debe desarrollar el sistema que en programar el desarrollo y las pruebas.
- **Metodologías ágiles:** Se centran en el software y no en su diseño y documentación. Fueron diseñadas principalmente para apoyar el desarrollo de aplicaciones de negocio, donde los requisitos cambian constantemente y están pensados para entregar de forma rápida y convencional el software a los clientes.

Después de un estudio realizado a las dos clasificaciones de metodologías existentes: ágiles y robustas, se determinó emplear para el desarrollo de esta herramienta, las metodologías ágiles, ya que son centradas en el software, están pensadas para aplicaciones que se deban entregar en un período de tiempo rápido y donde los requisitos cambian constantemente. Estas características son las que se ajustan al contexto de este problema.

1.4.1 Microsoft Solutions Framework (MSF)

Microsoft Solutions Framework es un marco de trabajo que se adapta de forma flexible a las características de cada proyecto. Con la aparición de la herramienta Microsoft Visual Studio Team System, se ha actualizado MSF a la versión 4.0, produciendo dos variantes: MSF para el Desarrollo de Software Ágil para el trabajo en entornos que emplean esta clasificación de metodología y MSF para CMMI (Capability Maturity Model for Integration) para la mejora de procesos. MSF Ágil está compuesta por las siguientes fases:

Visión y Alcance[2]: En esta fase se definen los objetivos que se persiguen y hasta dónde se quiere llegar con el proyecto. Este proceso se documenta en la declaración de visión que no es más que una descripción del valor del producto que será construido.

- **Planificación:** Es en esta fase cuando se termina la planificación del proyecto, el equipo prepara las especificaciones funcionales, se realiza el proceso de diseño de la solución y se preparan los planes de trabajo, estimaciones de costo y cronogramas de los diferentes entregables del proyecto.
- **Desarrollo:** Durante esta fase el equipo realiza la mayor parte de la construcción de los componentes (tanto documentación como código) y se crea una solución de la arquitectura a establecer. Como resultado se

[2] Figueroa, Robert G. y Solis, Camilo J. s.f. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.*

obtiene una versión de la infraestructura del producto después de aplicarle revisiones de código que se va generando.

- **Estabilización:** En esta fase se llevan a cabo las pruebas sobre la solución, que enfatizan el uso y operación bajo condiciones realistas. El equipo se enfoca en detectar los errores, solucionarlos y preparar la solución para el lanzamiento.
- **Implantación o Despliegue:** En esta fase se decide la tecnología base y sus componentes, estabiliza la instalación y se traspasa el proyecto al cliente.

Estos modelos de MSF tienen seis roles que corresponden a las metas de un proyecto y éstos son responsables de que las mismas sean cumplidas. Todos los roles de un proyecto tienen igual importancia en su aporte al mismo, mostrando que no es un modelo jerárquico, aunque éstos pueden tener diferentes niveles de actividades durante las etapas de desarrollo y ninguno puede ser omitido.

MSF cuenta además con modelos que se encargan de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión de Riesgos, Modelo de Diseño de Soluciones, Modelo de Infraestructura, Modelo de Costo Total de Propiedad y finalmente el Modelo de Aplicación.

Teniendo en cuenta lo anterior expuesto es que se define esta metodología de desarrollo ágil para desarrollar la propuesta que presenta este trabajo. Esto se debe a que es la metodología creada por Microsoft para los softwares que son desarrollados en la plataforma .NET. Además, está enfocada a dirigir proyectos o

soluciones de innovación y se centra en la gestión y administración del proyecto para lograr el impacto deseado.

1.5 Plataforma de desarrollo

Una plataforma de desarrollo es el entorno de software en el cual se desenvuelve la programación de una aplicación. Para la realización de la herramienta de apoyo al modelado del estudiante, se propone la utilización de los siguientes elementos que conforman el entorno de desarrollo.

1.5.1 Lenguaje de Modelado

El lenguaje de modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. Se usan en combinación a la metodología de desarrollo de software a emplear (BOOCH 1999).

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software muy utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables (Sommerville 2005).

- **NET Framework 4**

.NET Framework es el modelo de programación completo y coherente de Microsoft para compilar aplicaciones que ofrezcan una gran experiencia visual al

usuario, comunicación perfecta y segura, además de la capacidad de modelar una amplia gama de procesos empresariales.

- **Lenguaje de Programación**

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos una computadora debe operar, cómo deben ser estos almacenados, transmitidos y qué acciones se deben tomar bajo una variada gama de circunstancias. Una característica relevante de los lenguajes de programación es precisamente que más de un programador puede tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa. Para el desarrollo de la herramienta HCC se emplea el lenguaje de programación C#, por las siguientes razones (Stroustrup, BjarneBjarne Stroustrup. El lenguaje de programación C++. Addison Wesley, 1998).

C# presenta considerables mejoras e innovaciones en áreas como seguridad de tipos, control de versiones, eventos y recolección de elementos no utilizados (liberación de memoria). Tiene la característica de estar comprobando que efectivamente los tipos de datos que se estén manejando correspondan a los validados para las funciones que han sido creadas; así también vigila que no se produzcan errores en operaciones matemáticas, además de que también impide el uso de variables que no han sido inicializadas. Todo esto permite que no se produzcan errores en el momento de la ejecución (Abundiz, 2009).

A continuación, se describen las principales características de C#. Alguna de las cuales no son exactamente propias del lenguaje, sino de la plataforma .NET en general.

a. **Uso de operadores:** el ambiente de trabajo es muy cómodo ya que tiene un ambiente amigable y clásico de las aplicaciones de Windows. En cuanto a la forma de programar, será fácil de usar para quien está familiarizado con C++, ya que su estructuración básica es muy similar, sin embargo C# detecta la creación de funciones complejas desde cero y declaración de variables globales (Abundiz, 2009).
b. **Facilidad de uso:** la propia sintaxis de C# incluye elementos propios del diseño de componentes. Es decir, la sintaxis de C# permite definir cómodamente propiedades, eventos o atributos (Abundiz, 2009).
c. **Administración de memoria:** C# tiene la característica de inicializar los datos o variables declaradas en el programa, además de que también de forma automática libera la memoria cuando el mismo programa lo cree conveniente. Es decir, tiene constructores y destructores, y estos actúan automáticamente a menos que se manipulen desde el código (Abundiz, 2009).
d. **Seguridad de Tipos**: todos los tipos de datos que se definan siempre se derivarán, incluso de forma implícita, de una clase base común llamada System.Object, por lo que dispondrán de todos los miembros definidos en ésta clase (Abundiz, 2009).

- **IDE Microsoft Visual Studio 2015**

Microsoft Visual Studio no es más que un entorno de desarrollo integrado (IDE) para Windows. Este entorno de desarrollo soporta diferentes lenguajes de programación entre los cuales se pueden mencionar Visual C++, Visual C#, Visual J#, y Visual Basic. Visual Studio 2015 es la solución de desarrollo de vanguardia que permite a los equipos de todos los tamaños diseñar y crear aplicaciones atractivas del gusto de los usuarios.

Las herramientas de planeación ágiles y flexibles, como planeación de la capacidad, paneles de tareas y administración de trabajos pendientes, le permiten usar técnicas de desarrollo incremental y metodologías ágiles a su propio ritmo. Posee disímiles características, entre las principales se encuentran las siguientes:

a. **Depuración y Diagnóstico**: la suite presenta IntelliTrace, que no es más que una valiosa característica de depuración. Los evaluadores pueden archivar errores enriquecidos y modificables para que los desarrolladores puedan reproducir siempre el error del que se informe y el estado en el que se encontró.
b. **Herramientas de Prueba:** Visual Studio 2015 incorpora todas las herramientas avanzadas de Microsoft para pruebas.
c. **Arquitectura y Modelado:** los diagramas por capas ayudan a garantizar el cumplimiento de la arquitectura y permiten validar artefactos de código con respecto a los diagramas. Además, dicha suite admite los cinco diagramas de UML más comunes que conviven junto con su código.

- **Microsoft Project Professional 2007**[3]

Microsoft Project Professional 2007 ayuda a simplificar la planeación, la colaboración y la administración de recursos de forma eficaz y visualmente mejorada, para que pueda abordar correctamente todo tipo de proyectos. Con Project Professional 2007, las personas, los equipos y la empresa pueden potenciar los resultados de una administración unificada de proyectos, agregando Microsoft Project 2007. Project 2007 garantiza que las organizaciones seleccionen y entreguen los proyectos correctos obteniendo a la vez mayor visibilidad y control sobre los recursos, lo cual redunda en una mejor productividad y un mejor rendimiento de la empresa.

1.6 Conclusiones del capítulo

A partir de la literatura científica revisada se puede concluir la factibilidad de la clasificación supervisada basada en los conceptos de las clases, los cuales garantizan que no existe objeto alguno en las restantes clases que puedan ser descritos por ellos. Por tanto, es posible asegurar con total certeza que si existe un concepto que describe al objeto a clasificar, éste se clasifica de manera correcta. Además, teniendo en cuenta las características de la herramienta computacional que se implementa se considera que la plataforma .NET con el lenguaje C# y su metodología MSF Ágil constituyen una opción viable para su desarrollo.

[3] ***MICROSOFT. Microsoft Download Center.*** *Microsoft Office Project 2007. [En línea] [Citado el: 15 de Noviembre de 2016.]* *http://www.microsoft.com/es-es/search/DownloadResults.aspx?q=microsoft%20office%20project%202007*.

CAPÍTULO 2. ACC: ALGORITMO DE CLASIFICACIÓN SUPERVISADA UTILIZANDO LOS CONCEPTOS DEL AGRUPAMIENTO CONCEPTUAL

En este capítulo se describen los conceptos básicos del agrupamiento conceptual el cual es el fundamento teórico del algoritmo ACC que se propone como resultado de este trabajo. Se realiza una validación del algoritmo ACC dirigida a la eficacia en la clasificación.

2.1. Algoritmo LC-conceptual

El LC-conceptual (Pons-Porrata, Ruiz-Shulcloper y Martínez-Trinidad, 2002) es un algoritmo no supervisado, que retoma las ideas propuestas por Michalski (Michalski 1980) y está basado en los conceptos del Reconocimiento Lógico Combinatorio de Patrones. Basa su funcionamiento en dos etapas fundamentales: una de estructuración extensional donde se forman los grupos de objetos semejantes y otra de determinación intencional en la que se construyen los conceptos que caracterizan a los agrupamientos formados previamente.

Para describir cada una de las etapas se deben comprender con claridad los significados de l-complejo y de testor típico.

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, $\hat{i \in l}\,[x_i \# R_i]$, se denomina complejo lógico o l-complejo, siendo I, un conjunto de índices. 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.

De esta manera se puede afirmar que un l-complejo es una representación simbólica del conjunto de objetos que el cubre. Es importante subrayar además, que cualquiera de las regularidades discriminantes anteriormente mencionadas, se puede representar mediante l-complejos (Ruiz-Shulcloper 2009). Para la construcción de los l-complejos se utilizan los testores.

Definición 2.1: Sea T un subconjunto de rasgos de R ($T \subseteq R$) diremos que T es un testor respecto a la muestra de aprendizaje si sus rasgos son suficientes para diferenciar los elementos de cada clase con respecto a todos los demás de las restantes clases.

Definición 2.2: Si no existe $T' \subset T$ tal que T' no es testor entonces T es un testor típico.

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 éstos.

Ya construidos los agrupamientos se caracteriza cada uno de éstos utilizando para ello el operador de REFUNION (Pons-Porrata, Martínez-Trinidad, y Ruiz-Shulcloper, 2000), Condicionada RUC : $2^U \cup 2^{L(U)} \rightarrow 2^{L(U)}$ [2]. Este operador utiliza al conjunto de los testores típicos para construir propiedades o conceptos que satisfacen los objetos de cada agrupamiento.

El operador de REFUNION condicionada transforma un conjunto de objetos y/o l-complejos en un conjunto de l-complejos, y se define de la siguiente manera: RUC: $2^U U 2^{L(U)} \rightarrow 2^{L(U)}$.

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.

El procedimiento para realizar la REFUNION Condicionada es muy simple, el objetivo es construir para un conjunto de rasgos x_{i1} ,...,x_{ip}, un conjunto de valores $R_{ij,}$ j=1,...,p, asociado a cada uno de ellos, que son los que aparecerán en el selector del l-complejo.

Así que para un agrupamiento a caracterizar: se considera el primer objeto O_1, sin pérdida de generalidad, se supone que éste tiene la siguiente descripción O_1= ($x_{i1}(O_1)$,…, $x_{ip}(O_1)$), entonces al formar los conjuntos R_{ij}, éstos tendrán un solo valor, es decir, R_{i1}={$x_{i1}(O_1)$},…, R_{i1}={ $x_{i1}(O_1)$},…,R_{ip}={ $x_{ip}(O_1)$}. Después se considera que el segundo objeto es O_2= ($x_{i1}(O_2)$,…, $x_{ip}(O_2)$), y se agrega a R_{i1} el valor x_{i1} (O_2) sólo si éste, en combinación con valores del resto de los conjuntos, no aparece en el complemento del agrupamiento. En este caso si ($x_{i1}(O_2)$, $x_{i1}(O_1)$…, $x_{ip}(O_1)$) no aparece en algún objeto fuera del agrupamiento, entonces R_{i1} será R_{i1}= {$x_{i1}(O_1)$, $x_{i1}(O_2)$}. Este mismo proceso se repite para cada valor $x_{i1}(O_2)$ del segundo objeto O_2 .

En etapas posteriores del proceso se tendrán conjuntos más grandes R_{ij} y cada vez que se considere un valor $x_{ik}(O)$ para ser agregado a algún conjunto R_{ik} se tendrá que verificar que ninguna de las combinaciones (r_{i1},…,$x_{ik}(O)$, …,r_{ip}) donde r_{ij} $\in$ R_{ij} y $x_{ik}(O)$ es el valor analizado para ser o no incorporado a R_{ik}, aparezca en el

complemento del agrupamiento. El proceso antes descrito se aplica a todos los objetos del agrupamiento (Ruiz-Shulcloper 2009).

Sea U un universo de objetos, $R=\{x_1, \ldots, x_m\}$ el conjunto de rasgos en términos de los cuales se describen los objetos, con $x_i(O) \in M_i$, siendo M_i el conjunto de valores admisibles del rasgo x_i, además, se asume que en M_i existe un símbolo $*$, el cual denota la ausencia de información, i=1,…,m. Considérese $M_1 x \ldots x M_m = ERI$ (espacio de representación inicial). La naturaleza de estos rasgos será, simultáneamente, cualquiera (cualitativa: booleana, multivariada, difusa, lingüística y otras o cuantitativa: entera, real).

El problema sobre U consiste en determinar el conjunto cubrimiento $\{K_1, \ldots, Kc\}$, r>1 así como el conjunto de conceptos asociados a cada K_i i=1, …, c.

El **algoritmo LC-Conceptual** (Martínez-Trinidad 2001)**,** (Martínez Trinidad 1998) se describe a través de los siguientes pasos:

1 Determinar los criterios de comparación de las variables, la función de semejanza y el criterio de agrupamiento duro.

2 Estructurar la muestra inicial (MI) con el criterio de agrupamiento seleccionado.

3 Cada agrupamiento constituye una clase y se calculan todos los testores típicos para estas clases.

4 Para cada clase formar la estrella con todos los testores típicos, usando el operador de REFUNION CONDICIONADA (RUC).

5 Los conceptos que caracterizan a cada agrupamiento Ki son los obtenidos en el paso anterior como resultado de la aplicación del operador RUC.

El algoritmo LC-Conceptual manipula 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. Cada agrupamiento tiene asociado uno o más conceptos que lo representan, tantos como testores típicos sean seleccionados.

Este es un algoritmo que resuelve problemas de tipo no supervisados en el RLCP, no obstante, sus presupuestos teóricos pueden ser extendidos para la solución de problemas de clasificación de objetos, lo cual constituye el objetivo de esta investigación.

2.2. Algoritmo para el cálculo de los conceptos de las clases

El algoritmo de clasificación conceptual ACC que se propone está basado en la teoría del agrupamiento conceptual del RLCP, introduciendo como idea central clasificar un nuevo objeto a partir de la similitud con los conceptos que describen las clases.

Este modelo de clasificación conceptual es aplicable tanto a problemas no supervisados como supervisados. En el caso de los problemas no supervisados, es necesario previamente aplicar un algoritmo para obtener una estructuración de la muestra de objetos utilizada, basado en el grado de similitud de éstos, transformándolo así en un problema supervisado.

En este trabajo se asume que un problema supervisado se describe a través de una matriz de entrenamiento $K_1UK_2U\ldots K_r$, donde *r* representa el número de clases. La idea básica para el cálculo de los conceptos a partir de la etapa intencional del algoritmo LC-Conceptual se ilustra en el algoritmo 1.

Para facilitar la comprensión del algoritmo 1 se asume las definiciones que aparecen en de matriz de entrenamiento, matriz de diferencia y de matriz básica (Ruiz-Shulcloper 2009):

Definición 2.3: La **matriz de entrenamiento (**ME) describe un conjunto de objetos admisibles, donde $R= \{X_1, X_2, \ldots, X_n\}$ es el conjunto de rasgos en función de los cuales se describen los objetos admisibles de M.

Cada X_i tiene asociado un conjunto M_i denominado conjunto de valores admisibles del rasgo X_i, $i = 1, \ldots, n$ y un δ_i criterio de comparación asociado al rasgo i. $K_1, K_2, \ldots, K_r$ un cubrimiento finito de subconjuntos propios de M.

Definición 2.4: La matriz que se obtiene como resultado de comparar dos objetos de la matriz de aprendizaje que estén en clases distintas y que tendrá tantas filas como pares de objetos de tal tipo existan se le llama **matriz de diferencia** (MD) la cual tiene ***n*** columnas y *s* filas, de manera que si hay ***r***

clases y cada clase K_i tiene m_i elementos en la matriz de aprendizaje, entonces las s filas de la MD se calcula como:

$$s = \sum_{i=1}^{r} \sum_{j=i+1}^{r} m_i * m_j$$

Definición 2.5: se le llama **matriz básica** a la matriz formada exclusivamente por filas básicas de matriz de diferencia.

Definición 2.6: la **fila** i_t es **básica** sí y sólo sí en la matriz de diferencia no existe fila i_p alguna que sea subfila de i_t.

Definición 2.7: sea i_p, i_t filas de la matriz de diferencia. Diremos que i_p es **subfila** de i_t sí y sólo sí:

$$a) \forall j (a_{i_t j} = 0 \Rightarrow a_{i_p j} = 0$$
$$b) \exists j_0 (a_{i_t j_0} = 1 \wedge a_{i_p j_0} = 0$$

Algoritmo 1: Cálculo de los Conceptos por clases

Entrada: Matriz de Entrenamiento ($K_1 U K_2 U \ldots K_r$), donde *r* representa el número de clases.

Salida: Conceptos por clases (C_i).

Paso 1: Calcular el Conjunto de Testores.

a. Calcular la Matriz de Diferencias.

b. Calcular la Matriz Básica //representa el conjunto de testores

Paso 2: Calcular el conjunto de testores típicos clásicos (TT)// aplicar un algoritmo para el cálculo de los testores típicos BT,TB, CC, LEX, FastBR [15].

Paso 3: Calcular la utilidad de los testores típicos ψ.

a. Calcular la importancia de los rasgos que componen los testores según: la frecuencia de aparición (eq.6), la longitud de los testores (eq. 7) o ambas (eq.8 en la que α>O, β>O y α+β=1).

$$F(x_i) = \frac{|TT(x_i)|}{|TT|} \quad (6) \qquad L(x_i) = \frac{\sum_{t \in |TT(x_i)|} \frac{1}{|t|}}{|TT(x_i)|} \quad (7)$$

$$\varepsilon_i(x_i) = \alpha F_i(x_i) + \beta L_i(x_i) \quad (8)$$

donde |TT(xi)| número de testores donde aparece el rasgo i, |TT| número de testores, |ti| número de rasgos que forman el testor ti.

b. Calcular la utilidad de cada testor según eq. 9.

$$\psi_i(t_i) = \frac{\sum_{i=1}^{|t_i|} \varepsilon(x_i)}{|t_i|} \quad (9)$$

Paso 4: Ordenar descendentemente el conjunto de testores típicos a partir de la utilidad de cada t.

Paso 5: Calcular el concepto (Ci) de cada clase Ki, utilizando el testor típico de mayor utilidad.

Como resultado de este algoritmo se obtienen los conceptos que describen cada clase, fundamento del algoritmo de clasificación conceptual ACC. La complejidad temporal de este algoritmo es alta, exponencial en el peor de los casos y está determinada por el cálculo de los testores típicos y los conceptos, sin embargo, éste sólo se ejecuta en la etapa de ingeniería del conocimiento, y resulta de gran

utilidad pues constituye la base de la eficiencia y eficacia para el algoritmo que se propone.

2.3. Algoritmo Clasificación Conceptual (ACC)

La idea central del algoritmo ACC es clasificar un nuevo objeto calculando el grado de similitud con el concepto de cada clase, a partir de una función de semejanza que integra los rasgos que componen el concepto y la importancia de los mismos. Si el grado de semejanza del nuevo objeto con cada uno de los conceptos de las clases es menor que el umbral de semejanza establecido, el algoritmo se abstiene de clasificar. Por el contrario, clasifica el objeto en la clase con mayor valor de similitud.

Algoritmo 2: ACC Algoritmo de clasificación conceptual

Entrada: On, objeto nuevo a clasificar

K_i, i=1…r, con r número de clases

Cj, j=1…r , Conjuntos de conceptos por clases.

B_0, umbral de semejanza (por criterio de expertos o según ecuaciones del RLCP)

Γ, es una función de semejanza seleccionada según el dominio de aplicación.

Salida: K_i, clase a la que pertenece el objeto On.

Paso 1: Para cada clase calcular $\Gamma_i(O_n,C_i)$, donde Ci concepto de la clase i.

Paso 2: Seleccionar el $\Gamma_i=1$.

Paso 3: Si no existe $\Gamma_i=1$ entonces, se abstiene de clasificar.

2.4. Eficacia del algoritmo de clasificación conceptual ACC

Para validar la precisión de un modelo de clasificación no se debe utilizar el conjunto de entrenamiento, sino un conjunto de prueba, pues de lo contrario el resultado puede conllevar a un error de re-sustitución.

Existen distintas técnicas de validación de clasificadores, entre ellas:

- Hold - out
- Validación cruzada
- Leave – one - out
- Boostraping

En este trabajo se sigue la idea tomada de (González, Martínez, y Díaz, 2014) que se utiliza el método de validación cruzada (*cross validation en inglés*) el cual consiste en dividir la base de casos en *n* conjuntos de igual tamaño y efectuar el algoritmo de aprendizaje *n* veces, en cada una de las cuales el conjunto de entrenamiento o muestra de aprendizaje son todos, excepto uno de los n subconjuntos o muestra de control donde se evalúa.

Por características propias del algoritmo ACC no es aplicable esta técnica sin modificaciones. Como se explica en el capítulo 2 de este trabajo, los conceptos, fundamento teórico en el que se basa el algoritmo que se propone, se calculan a partir de los objetos que conforman el grupo, por lo que si no existe objeto alguno en el grupo con estas propiedades el concepto deja de ser concepto del grupo, lo que implica que al entrenar el clasificador, con el conjunto de objetos de la muestra seleccionada para la prueba, no necesariamente se generan los mismos conceptos y esto provoca un enmascaramiento de la eficacia del algoritmo ACC.

Por tal razón, en la validación experimental se sigue la idea de la técnica de validación cruzada de realizar *k* iteraciones de prueba sin entrenar el clasificador.

Tabla 2.1 Características de las bases de datos utilizadas en la validación

Base de casos	Cantidad de instancias	Cantidad de atributos	Cantidad de Clases	Ausencia de información
Breast-Cancer	**286**	**9**	**2**	**Sí**
Hepatitis	**155**	**19**	**2**	**Sí**
Labor	**57**	**16**	**2**	**No**
Postoperative-patient-data	**90**	**8**	**3**	**Sí**

El esquema de estudio experimental que se aplica es el siguiente:

a. Formación de muestras de entrenamiento y control.

Se obtienen los conjuntos de entrenamiento y control, siguiendo el principio de validación cruzada, se selecciona el 90% de los objetos para el entrenamiento y el 10%, para conjunto control.

b. Proceso de clasificación supervisada.

Se realiza el proceso de clasificación supervisada a través de la herramienta HCC, específicamente con los clasificadores supervisados Ideal de la clase, Holotipo de la clase y el algoritmo ACC.

Inicialmente se asume como un problema de clasificación no supervisada y se selecciona el mismo criterio de agrupamiento y umbral de semejanza, lo que permite trabajar con el mismo número de clases y éstas contienen los mismos objetos.

El umbral de semejanza se calcula utilizando la ecuación (Ruiz-Shulcloper 2009):

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

c. Análisis comparativo de los resultados.

Se comparan los por cientos de clasificaciones correctas en cada una de las 10 iteraciones realizadas. Además, se hace un análisis distintivo en la iteración 11 cuya muestra está conformada por los objetos no observados, que son aquellos objetos que no están físicamente en la clase, pero pueden ser calculado utilizando los conceptos de cada clase y por ende conocer a la clase que pertenecen.

d. Cálculo de las Medidas de Inferencia.

A partir de la idea estudiada en (Caballero 2006) y (Gonzalez, Martínez, y García, 2015) se utilizan las medidas de la Teoría de los Conjuntos Aproximados (Skowron 2005):

Calidad de la clasificación. Este coeficiente describe la inexactitud de las clasificaciones aproximadas:

$$\Gamma(DS) = \frac{\sum_{i=1}^{l} \left| R'_{*}(X_i) \right|}{\left| U \right|}$$

La medida calidad de la clasificación expresa la proporción de objetos que pueden estar correctamente clasificados en el sistema. Si ese coeficiente es igual a 1, entonces el sistema de decisión es consistente, en otro caso es inconsistente.

2.5. Resultados del estudio experimental

- Formación de muestras de entrenamiento y control.

Las figuras 2.1, y 2.3 ilustran como la herramienta HCC muestra los resultados de las pruebas en una de las iteraciones realizadas y la tabla 2.2 los resultados de las iteraciones.

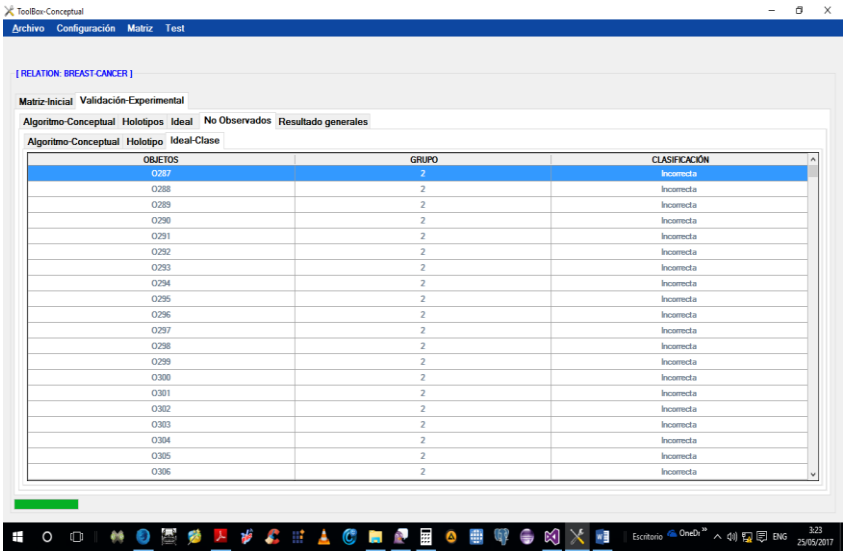


Figura 2.1. Clasificación utilizando el Ideal

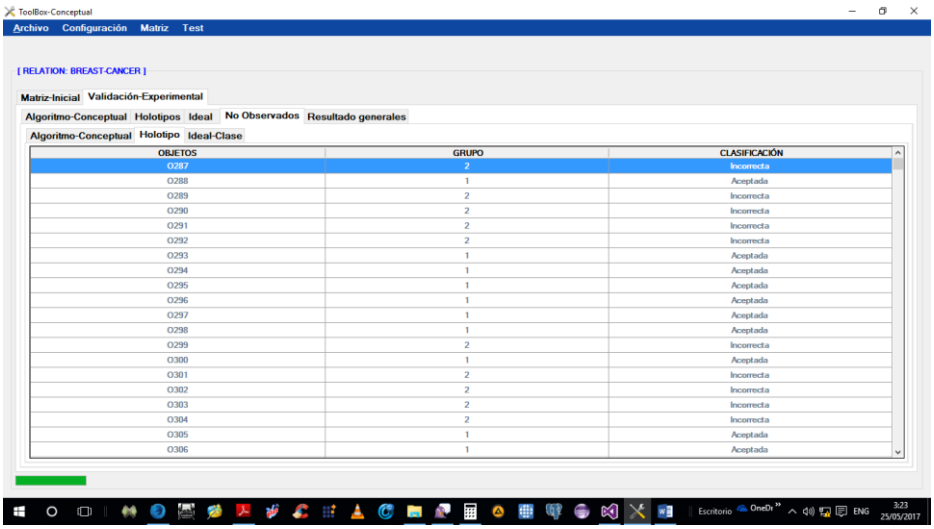


Figura 2.2. Clasificación utilizando el Holotipo

- Proceso de clasificación supervisada.

Tabla 2.2. Comportamiento de los algoritmos con las muestras seleccionadas

Base de Datos (BD)	|BD|	Clases	|Muestra|	$\beta 0$	IDEAL	HOLOTIPO	ACC
Breast -Cancer	286	14	28	0,83	32,5	28,7	100
Hepatitis	155	8	15	0,86	84,5	84,5	100
Labor	57	7	5	0,82	49,12	65	100
Postoperative-patient-data	90	11	9	0,83	40	9	100

Donde, |BD|: cantidad de objetos de la clase, |Muestra|: cantidad de objetos de cada muestra y β_0: umbral de semejanza para la clasificación.

- Análisis comparativo de los resultados.

La figura 2.3 ilustra el comportamiento de los tres algoritmos con las bases de datos utilizadas para la clasificación supervisada.

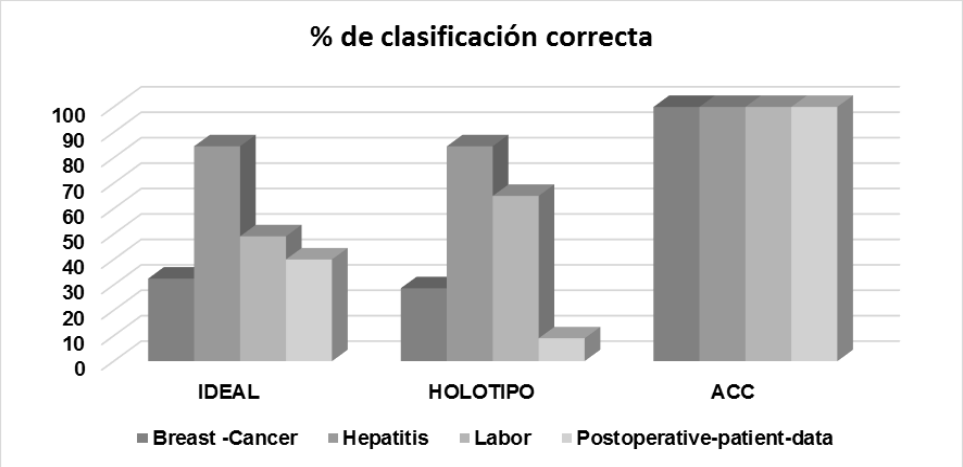


Figura 2.3. Resultados de la validación realizada

En los anexos se muestran el comportamiento de la validación para cada una de las iteraciones realizadas.

La siguiente figura 2.4 ilustra el comportamiento de la validación realizada utilizando los algoritmos Ideal y Holotipo, tomando como muestra los objetos no observados que se generan a través de los conceptos.

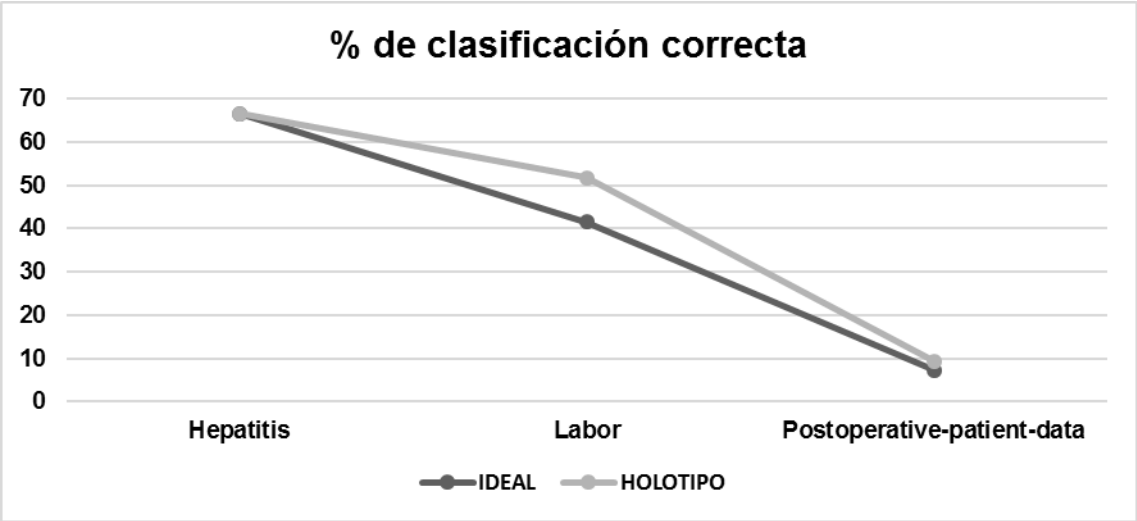


Figura 2.4. Validación de los objetos no observados

- Cálculo de las Medidas de Inferencia.

Para calcular la calidad de la clasificación es necesario obtener el número de objetos clasificados correctamente en cada iteración en todas las bases de datos utilizadas en la validación experimental. Las tablas 2.3 y 2.4 muestran estos resultados y los resultados totales de la clasificación correcta y la calidad de la clasificación de cada algoritmo respectivamente.

Tabla 2.3. Clasificación correcta por algoritmos

Base de Datos (BD)	|BD|	IDEAL	HOLOTIPO	ACC
Breast -Cancer	286	93	82	286
Hepatitis	155	131	131	155
Labor	57	28	37	57
Postoperative-patient-data	90	36	8	90
UNIVERSO	588	288	258	588

Tabla 2.4. Calidad de la clasificación utilizando

ALGORITMOS	IDEAL	HOLOTIPO	ACC
$\gamma(DS) = \frac{\sum_{i=1}^{c} \lvert R^i \cdot (X_i) \rvert}{\lvert U \rvert}$	49%	44%	100%

La validación experimental muestra resultados sustantivamente superiores del algoritmo ACC con respecto a los otros dos restantes. Independientemente que se puede garantizar la correcta clasificación con el algoritmo ACC los resultados, no se puede afirmar categóricamente esta tendencia, los mismos están estrechamente relacionados con las características de las bases de datos utilizadas, el criterio de agrupamiento seleccionado, el umbral de semejanza, entre otros parámetros a tener en cuenta.

2.6. Conclusiones del capítulo

En este capítulo se formaliza el algoritmo de clasificación conceptual ACC para problemas de clasificación supervisados. Se definen los conceptos del RLCP necesarios para la fundamentación teórica del algoritmo.

Se describe el esquema de estudio experimental que se aplica y se realiza un análisis crítico de los resultados el cual arroja que la calidad de la clasificación del algoritmo ACC es superior a la de los algoritmos del RLCP basados en el Ideal de la clase y el Holotipo de la clase para las bases de datos utilizadas.

El algoritmo ACC por la característica distintiva de los conceptos del agrupamiento conceptual garantiza que la calidad de la clasificación tome el valor máximo.

CAPÍTULO 3. REALIZACIÓN COMPUTACIONAL DEL ALGORITMO ACC

El algoritmo propuesto en el capítulo 2 constituye el fundamento para el diseño y la implementación software HCC: herramienta para la clasificación supervisada utilizando el agrupamiento conceptual. Se exponen las características generales de ésta, así como los requerimientos técnicos necesarios para su implantación. La descripción de un caso de estudio y su validación utilizando el esquema experimental descrito en el capítulo 2 son tratados en los últimos epígrafes.

3.1. Herramienta Computacional para la clasificación supervisada utilizando los conceptos: HCC

Para el desarrollo de la herramienta computacional HCC se utilizó la metodología de software MSF Ágil, la cual facilita el conocimiento de las funcionalidades de la herramienta y organiza el proceso de desarrollo del software en iteraciones con el fin de culminar ésta en el tiempo requerido.

La arquitectura de software empleada es de N-Capas representada a través de cinco capas lógicas, brinda un alto nivel de encapsulamiento de las responsabilidades, permitiendo reducir al máximo el acoplamiento y así aumentar la reutilización entre las mismas. Los patrones de diseños utilizados facilitan un paquete de código altamente cohesivo, con responsabilidades bien definidas y relacionadas, además permiten que las capas solo tengan visibilidad a los recursos necesarios de otras capas para su funcionamiento.

Las estrategias de pruebas seguidas: las pruebas unitarias que se le realizaron demostraron la factibilidad y el buen funcionamiento de la herramienta

computacional HCC desarrollada. A continuación, se muestra un ejemplo de pruebas unitarias realizadas a unos de los métodos.

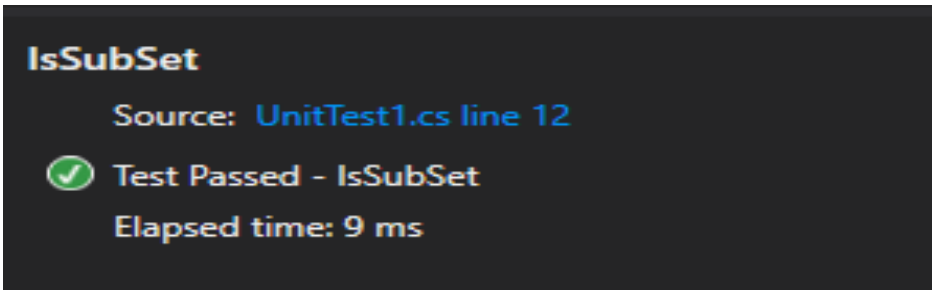
Prueba de Unidad		
Nombre de prueba: IsSubSet		
Estado: Satisfactoria	**Tipo:** Caja blanca	**Ultima ejecución:20/06/2017**
Ejecutado por: Yoandy Figuerola	**Verificado por:** Yunia Reyes Gonzáles	
Descripción: Para la ejecución de esta prueba se debe entrar dos conjuntos de elementos **a** y **b** para determinar si el primer conjunto **a** es subconjunto del segundo conjunto **b**.		
Entrada: Conjunto a, Conjunto b		
Criterio de aceptación: Devolver verdadero o falso		
		

Tabla 2.5: Descripción de la prueba IsSubset.

Prueba de Unidad		
Nombre de prueba: IsSuperset		
Estado: Satisfactoria	**Tipo:** Caja blanca	**Ultima ejecución:20/06/2017**
Ejecutado por: Yoandy Figuerola	**Verificado por:** Yunia Reyes Gonzáles	
Descripción: Para la ejecución de esta prueba se introducen dos conjuntos de elementos **a** y **b** para determinar si el conjunto **a** es superconjunto de **b**.		

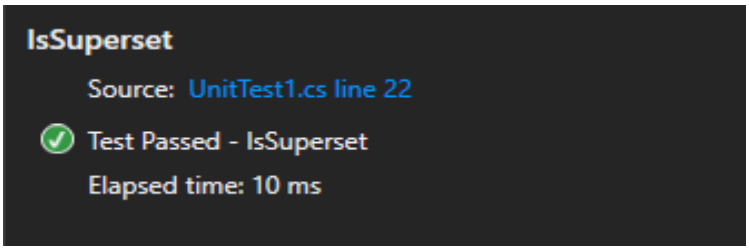
Entrada: Conjunto a, Conjunto b
Criterio de aceptación: Devolver verdadero o falso


Tabla 2.6: Descripción de la prueba IsSuperset.

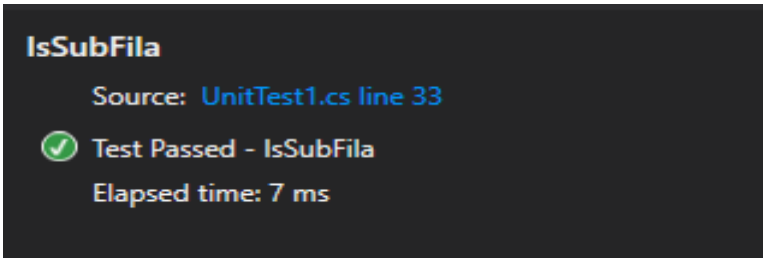
Prueba de Unidad		
Nombre de prueba: IsSubFila		
Estado: Satisfactoria	**Tipo:** Caja blanca	**Ultima ejecución:20/06/2017**
Ejecutado por: Yoandy Figuerola	**Verificado por:** Yunia Reyes Gonzáles	
Descripción: Para la ejecución de esta prueba se introducen dos conjuntos de elementos **a** y **b** respectivamente, para determinar si el primero es subfila del segundo.		
Entrada: Conjunto a, Conjunto b		
Criterio de aceptación: Devolver verdadero o falso		
		

Tabla 2.7: Descripción de la prueba IsSubFila.

La herramienta computacional HCC ofrece de funcionalidades útiles a los investigadores de esta rama del saber. Cuenta con implementaciones de algoritmos del RLCP que responden a los cuatros problemas fundamentales a los que se dedica esta área del saber; clasificación supervisada, no supervisada, parcialmente supervisada y a la selección de rasgos y el cálculo de la importancia de los mismos. La figura 3.1 ilustra una de las ventanas iniciales de la herramienta.

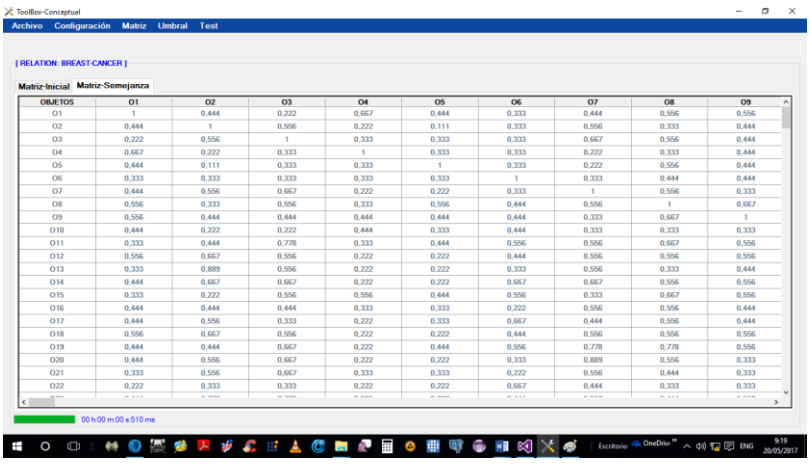


Figura 3.1. Matriz Inicial

Los algoritmos de clasificación no supervisada permiten utilizar varios de los tipos de agrupamientos definidos en la literatura, así como el cálculo del umbral de semejanza por diferentes vías. La figura 3.2 ilustra el ejemplo de una ventana que permite seleccionar el método que se desea para el cálculo del umbral de semejanza.

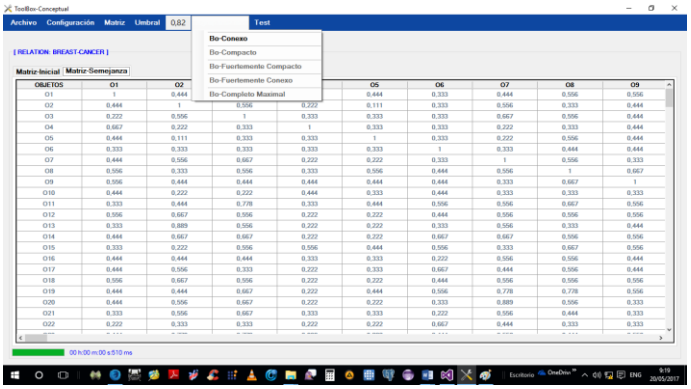


Figura 3.2. Umbral de semejanza

En la figura 3.3 se ilustra el menú principal donde se puede apreciar la utilidad de la herramienta computacional HCC para los investigadores. A partir de una matriz inicial o base de datos se puede agrupar los objetos por diferentes criterios, calcular los rasgos relevantes utilizando diferentes algoritmos para el cálculo de los testores típicos, y los conceptos de los grupos, esencia del algoritmo de clasificación conceptual ACC.

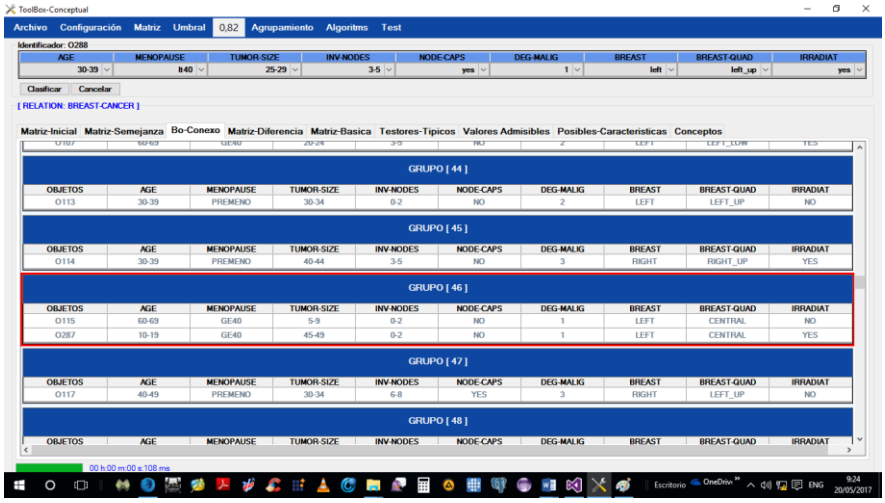


Figura 3.3. Herramienta HCC

Además, la herramienta computacional HCC facilita la comparación de los algoritmos de clasificación supervisada Ideal, Holotipo y ACC, lo cual le da un valor agregado a ésta. La figura 3.4 ilustra una ventana con estas funcionalidades.

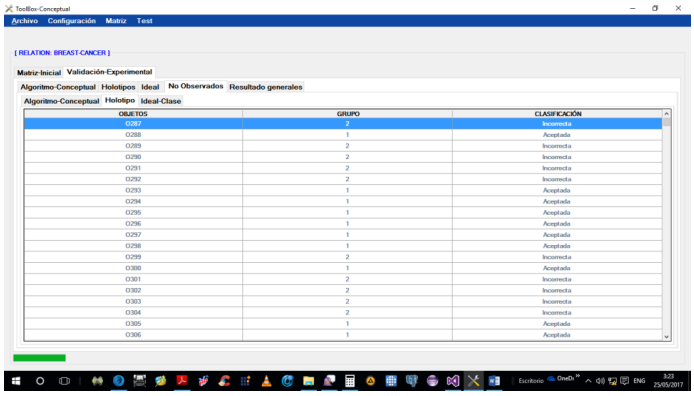


Figura 3.4. Algoritmos de clasificación supervisada

3.2. Ejemplo de un caso de estudio utilizando la herramienta HCC

Para resolver un problema de reconocimiento de patrones, se necesita ante todo modelar y después procesar los datos. En muchos casos la solución final es un programa computacional que el usuario emplea para resolver la dificultad en cuestión. Esto significa que la solución de un problema de reconocimiento de patrones implica: MODELACIÓN → PROCESAMIENTO DE DATOS → SOLUCIÓN

En la práctica casi siempre se tienen datos que necesitan de un proceso previo para poder extraer la información que se requiere. En el caso particular del reconocimiento de patrones, aunque no hay una frontera clara en el proceso a partir de los datos en bruto hasta las conclusiones finales, un paradigma útil es considerarlo dividido en cuatro tipos de procesos de datos: adquisición, pre procesamiento, representación/descripción y análisis. La etapa de adquisición es el primer paso para procesar datos. De hecho, la forma en que se adquieran los datos es la primera transformación que se les está haciendo, es decir, la forma en que sean adquiridos los datos afectará todo el proceso posterior. Esta etapa se caracteriza por el hecho que la entrada son los datos originales tomados de las

fuentes originales y la salida son los datos en bruto a partir de los cuales se debe extraer la información que se busca.

La etapa de pre procesamiento es un proceso de datos caracterizado por el hecho que tanto la entrada como la salida son datos de la misma naturaleza y significan lo mismo. La etapa de representación/descripción es el proceso en el cual los datos originales son transformados en una forma nueva, adecuada para el proceso posterior. Está caracterizada la misma por el hecho de que la entrada es diferente a la salida, al menos en su significado. Es un proceso por el que se describen los objetos involucrados en el problema. Finalmente, la etapa del análisis es un proceso en el cual se encuentra el significado de los datos originales o al menos de una parte de ellos. Se puede reconocer la ocurrencia de cierta información previamente almacenada y se puede tomar una decisión, llegar a una conclusión.

En este epígrafe se describe el caso de estudio utilizando la metodología que se propone (Ruiz-Shulcloper 2009), (Ruiz-Shulcloper, Carrasco-Ochoa, y Martínez-Trinidad, 2013) la cual consta de cinco etapas:

1. Formulación del problema no matemático, es decir el problema que se quiere resolver.
2. Formalización del problema, es decir, creación del problema matemático.
3. Selección de la forma de solución del problema.
4. Solución del problema matemático.
5. Análisis e interpretación de los resultados, respecto al problema.

Formulación del problema:

En el plan de estudios de la carrera de Ingeniería en Ciencias Informática el currículo optativo está conformado por ocho asignaturas distribuidas en los tres últimos años académicos. Estas asignaturas están vinculadas a seis de las 14 disciplinas que conforman el plan de estudios.

En la universidad se estableció un proceso que le permite a los estudiantes optar por cinco de las asignaturas que se ofertan en orden de prioridad y a partir de la demanda y la cantidad de plazas ofertadas teniendo en cuenta las variables (rasgos) que se muestra en la tabla 1 se le otorgan dos de las asignaturas solicitadas, las cuales deben cursar. Esta información se está almacenando en los últimos tres años con el objetivo de extraer conocimiento que permita perfeccionar este proceso.

Formalización del problema:

El universo se corresponde con la información almacenada relacionada con el problema formulado, donde un objeto equivale a un estudiante, los cuales se describen a través de 20 rasgos que se muestran en la tabla 3.1, así como su dominio y función de comparación.

La importancia de los rasgos se calcula utilizando los testores típicos basada en la frecuencia de aparición del rasgo en los testores típicos, teniendo en cuenta la longitud del testor al que pertenece el rasgo o ambas variantes. Estas son funcionalidades que ofrece la herramienta computacional HCC.

Tabla 3.1. Descripción de los rasgos de la base datos AO

Rasgo	Descripción	Dominio
Facultad	Facultad a que pertenece el estudiante	1,2,3,4, CITEC
Año	Año académico	3,4,5
Índice académico	Índice académico	[2,5]
OP_1, OP_2, OP_3, OP_4, OP_5,	Asignatura optativa según prioridad	ADA, RP, RNA, LP, entre otras.
D_1, D_2, D_3, D_4, D_5,D_6	Disciplina a la que pertenece la asignatura OP	IA,SD, IGSW,P, CE, MA
A,P,IGSW,CE,MA,SD	Disciplina a la que pertenecen las asignaturas cursadas en los semestres anteriores	1,2,3,4,5,6

Selección de la forma de solución del problema:

El problema formulado se corresponde con un problema de clasificación no supervisada, pues se conocen los objetos, pero no el número de grupo que se pueden formar a partir de la semejanza existente entre éstos, ni las características que los distingue.

Por lo que se selecciona realizar un agrupamiento conceptual, el cual en su etapa extensional permite estructurar la base de datos y en la etapa intencional determinar las características distintivas de cada grupo.

Se aplica el criterio agrupacional β_0 compacto y se calcula el umbral de semejanza utilizando la ecuación (Ruiz-Shulcloper 2009):

$$\beta_o = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \beta\left(O_i, O_j\right)$$

Solución del problema matemático

A partir de esta estructuración de la base de datos AO utilizando la etapa intencional del algoritmo LC-conceptual (Pons-Porrata, Ruiz-Shulcloper y Martínez-Trinidad, 2002), (Pons Porrata 2004), se calculan los conceptos de cada clase. La tabla 3.2 muestra los resultados de la validación realizada al caso de estudio descrito siguiendo el esquema de estudio experimental utilizado para la validación del algoritmo de clasificación conceptual ACC.

Tabla 3.2. Resultados de la validación del caso de estudio realizado

	Asignaturas-Optativas: 652	Muestra: 65		Umbral: 0,72		
	Algoritmos					
	Ideal		Holotipo		ACC	
Iteración	Correctamente	%	Correctamente	%	Correctamente	%
k=1	18	27,69	16	24,62	65	100,00
k=2	8	12,31	7	10,77	65	100,00
k=3	9	13,85	9	13,85	65	100,00
k=4	2	3,08	1	1,54	65	100,00
k=5	6	9,23	3	4,62	65	100,00
k=6	16	24,62	7	10,77	65	100,00
k=7	13	20,00	12	18,46	65	100,00
k=8	14	21,54	8	12,31	65	100,00
k=9	3	4,62	3	4,62	65	100,00
k=10	7	10,45	1	1,49	67	100,00
Total	**96**	**14,72**	**67**	**10,28**	**652**	**100,00**

Análisis e interpretación de los resultados, respecto al problema.

Los resultados que se obtienen de calcular los conceptos a cada uno de las clases que se forman permite realizar un estudio cualitativo de las características y tendencias de los estudiantes de la carrera de Ingeniería en Ciencias Informáticas en la selección de las asignaturas que como parte del currículo optativo del Plan de Estudio para complementar su formación profesional. Por otra parte, como resultado práctico de este trabajo, el colectivo de la carrera y los directivos docentes cuentan con una herramienta computación que le permite realizar diferentes estudios de apoyo a la toma de decisiones en cuanto al diseño de las asignaturas que se ofertan, estrategia de orientación profesional, entre otros aspectos.

3.3. Conclusiones del capítulo

En este capítulo se describen las características esenciales de la herramienta computacional para la clasificación supervisada utilizando los conceptos: HCC, la cual tiene como fundamento teórico el algoritmo de clasificación conceptual ACC propuesto como resultado de este trabajo.

La herramienta HCC con el objetivo de poder utilizar el algoritmo ACC tanto para problemas supervisados como no supervisado tiene implementado algoritmos de agrupamiento no supervisado, basados fundamentalmente en diferentes tipos de agrupamiento y cálculo del umbral de semejanza. Además, permite el cálculo de los testores típicos (Pons 2003) y de los conceptos de las clases utilizando el algoritmo LC-conceptual.

Por otra parte, con el objetivo de validar el algoritmo ACC se implementan los algoritmos de clasificación supervisada basado en el Ideal de la clase y basado en el Holotipo de la clase.

Además, se presenta un caso de estudio a solicitud de la Dirección Docente Metodológica de la universidad para analizar de manera cualitativa y no cuantitativa la tendencia de los estudiantes en la selección de las asignaturas optativas. Los resultados obtenidos son satisfactorios y de utilidad para el propósito solicitado.

CONCLUSIONES

El trabajo realizado responde a la línea de investigación de Inteligencia Artificial y Reconocimiento de Patrones de la Universidad de las Ciencias Informáticas.

Permite integrar los conocimientos y habilidades adquiridas en las 14 disciplinas que conforman el plan de estudios de la carrera de Ingeniería en Ciencias Informáticas con mayor énfasis, en las disciplinas de Inteligencia Artificial, Técnicas de Programación e Ingeniería y Gestión de Software, entre otras.

Como resultado de la investigación se propone un nuevo algoritmo de clasificación supervisada ACC utilizando los conceptos del agrupamiento conceptual, el cual la validación experimentar realizada, así como la comparación con dos de los algoritmos tradicionales del RLCP demuestra su eficacia.

La implementación del algoritmo de clasificación conceptual ACC en la herramienta computacional HCC aporta un valor práctico a los resultados alcanzados, puesto que en la herramienta se implementan algoritmos (Holotipo, IDEAL, ACC) del RLCP que permiten tanto el agrupamiento no supervisado a partir de diferentes criterios de agrupamiento, el cálculo de los rasgos relevantes y su importancia utilizando varios algoritmos del cálculo de los testores típicos, así como el cálculo de los conceptos, la clasificación conceptual y su comparación con dos de los algoritmos clásicos del RLCP.

Teniendo en cuenta, que la investigación realizada se desarrolla en el marco del RLCP, el cual es utilizado en aquellas ciencias como la Medicina, Sociología, Geociencias, Criminalística, donde frecuentemente se presentan problemas de clasificación, de diagnóstico, de pronóstico, de determinación de factores de

influencias, que hoy son de las principales líneas de desarrollo de los softwares que se implementan en los centros de la UCI, la utilización de los resultados alcanzados tienen un alto nivel de aplicabilidad, lo cual contribuye a la misión de la universidad de producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, así como, servir de soporte a la industria cubana de la informática.

RECOMENDACIONES

Derivadas del estudio realizado, así como de las conclusiones generales emanadas del mismo, se **recomienda**:

1. Implementar el algoritmo de clasificación supervisada KORA 3 que se basa en los rasgos complejos, para compararlo con el algoritmo propuesto.
2. Utilizar el algoritmo de clasificación conceptual ACC en los softwares que se desarrollan en la red de centros de la universidad, según su pertinencia.

REFERENCIAS BIBLIOGRÁFICAS

Ayaquica-Martínez, , J.F. Martínez-Trinidad, y J.A. Carrasco-Ochoa. 2005. Conceptual K-Means Algorithm with Similarity Functions. CIARP 2005, LNCS 3773, pp. 368 – 376, 2005. © Springer-Verlag Berlin Heidelberg 2005.

BOOCH, Grady, et al. 1999. El lenguaje unificado de modelado. . Editado por Addison-Wesley.

Caballero, Y. y Bello, R. (2006). 2006. «Two new feature selection algorithms with Rough Sets Theory.» (Artificial Intelligence in Theory and Practice. M. Bramer, Springer Boston. 217: 209-216.).

Cover, T. M. and P. E. Hart. 1967. "Nearest neighbour pattern classification." . Institute of Electronical and Electronics Engineers Transactions on Information Theory 13: 21-27.

Figueroa, Robert G. y Solis, Camilo J. s.f. s.f. METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. . Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación. .

González Reyes, Y, , N. Martínez Sánchez, y , A. Díaz Sardiñas. 2014. Modelo para la adaptación de las soluciones en un Sistema Basado en Casos utilizando el agrupamiento conceptual. La Habana, Cuba.: Tesis presentada en opción al Título de Máster en Informática Aplicada. Facultad 2, UCI.

González Reyes, Y, y N. Martínez Sánchez. 2016. Agrupamiento Conceptual Lógico- Combinatorio: una alternative para la toma de decisiones. Revista Iberoamericana de Inteligencia Artificial, [S.l.], v. 19, n. 57, p. 82-96, may. 2016. ISSN 1988-3064.

González Reyes, Y., y , N. Martínez Sánchez. 2014. «La toma de decisiones en los Sistemas Tutoriales Inteligentes utilizando el agrupamiento conceptual.» Revista Cubana de Ciencias Informáticas Vol. 8, No. Especial UCIENCIA 2014, Mayo, 2014 ISSN: 2227-1899 | RNPS: 2301 Pág. 104-116 http://rcci.uci.cu.

González, Y, N Martínez, , y Adolfo Díaz,. 2014. Modelo para la adaptación de las soluciones en un Sistema Basado en Casos utilizando el agrupamiento conceptual. Tesis presentada en opción al título de máster en Informaática Aplicada. Universidad de las Ciencias Informáticas. La Habana. Cuba.

Gonzalez, Y, N Martínez,, y MM García,. 2015. «EL AGRUPAMIENTO CONCEPTUAL EN EL CONTEXTO DE LA TEORÍA DE LOS CONJUNTOS APROXIMADOS.» DYNA New technologies. http://dx.doi.org/10.6036/NT7594 Volumen 2, p. 1-12.

Guerra-Gandón, A., S. Vega-Pons, , y J. Ruiz-Shulcloper,. 2012. Algoritmos de agrupamiento conceptuales: un estado del arte. Reporte Técnico de reconocimiento de Patones. RNPS no. 2142. ISSN 2072-6287. Serie Azul. CENATAV. la Habna Cuba.

Martínez Sánchez, N, Z Z. García Valdivia, , y MM. García Lorenzo,. 2009. Modelo para diseñar Sistemas de Enseñanza-Aprendizaje Inteligentes utilizando el Razonamiento Basado en Casos. Tesis presentada en opción del grado científico de Doctor en Ciencias Técnicas. Facultad de Matemática, Física y Computación, UCLV, Villa Clara, Cuba.

Martínez Trinidad, J. 1998. “Fuzzy LC-Conceptual Algorithm. Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing, Aache, Germany, September 7-10, 1998, Vol. 1, pp. 20-24. ." .

Martínez-Trinidad, J. F. a. G. S.-D. 2001. "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. 117127.". 58.

Martínez-Trinidad, JF., y J. Ruiz-Shulcloper,. 1998. «FUZZY LC-CONCEPTUAL ALGORITHM.» (In: Proceedings 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT 1998)).

Martnez-Trinidad, JF., y G. Sánchez-Díaz,. 2001. «LC: A Conceptual Clustering Algorithm.» P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 117127, 2001. ' Springer-Verlag Berlin Heidelberg 2001.

Michalski, R. S. 1980. "Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. Journal of Policy Analysis and Information Systems, 1980, vol. 4, no 3, p. 219-244.

Pons Porrata, A. 2004. "DESARROLLO DE ALGORITMOS PARA LA ESTRUCTURACIÓN DINÁMICA DE INFORMACIÓN Y SU APLICACIÓN A LA DETECCIÓN DE SUCESOS. . Trabajo de Tesis en opción al grado científico de Doctor en Ciencias Técnicas." .

Pons, A. 2003. "LEX: Un nuevo algoritmo para el cálculo de los testores típicos. La Habana: Revista Ciencias Matemáticas. Cuba.".

Pons-Porrata, A., JF. Martínez-Trinidad,, y J. Ruiz-Shulcloper,. 2000. «REFUNION-GENERALIZATION-CONCEPTUAL CLUSTERING ALGORITHM.» V Simposio Iberoamericano de reconocimiento de Patrones. Portugal. 679-690.

Pons-Porrata, A.,, J. Ruiz-Shulcloper , y J.F. Martínez-Trinidad,. 2002. RGC: a new conceptual clustering algorithm for mixed incomplete data sets. In Mathematical and Computer Modelling. p. 1375-1385.".

Rodríguez Sarabia, Y, MM García Lorenzo,, y B. De Baets, . 2007. 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. Villa Clara Cuba.

Ruiz-Shulcloper, J. 2009. RECONOCIMIENTO LÓGICO COMBINATORIO DE PATRONES: TEORÍA Y APLICACIONES. . TESIS EN OPCIÓN AL GRADO CIENTÍFICO DE DOCTOR EN CIENCIAS.Centro de Aplicaciones de Tecnologías de Avanzada. .

Ruiz-Shulcloper, J., JA. Carrasco-Ochoa, , y JF. Martínez-Trinidad,. 2013. Reconocimiento de patrones: conceptos y metodología. La Habna. Cuba: Reporte Técnico de Reconocimiento de Patrones. RNPS no. 2142, ISSN 2072-6287. Serie Azul RT 054. CENATAV.

Skowron, J. F. Peters and A. 2005. «Rough Sets and Bayes Factor. Transactions on Rough Sets III. .» Lecture Notes in Computer Science. Springer-Verlag GmbH. 3400.

Sommerville, Ian. 2005. Ingeniería del software. Pearson Educación.

Burget, R., Karasek, J., Smekal, Z., Uher, V., & Dostal, O. (2010). Rapidminer image processing extension: A platform for collaborative research. In The 33rd International Conference on Telecommunication and Signal Processing, TSP 2010 (pp. 114-118).

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing, 17.

Stroustrup, BjarneBjarne Stroustrup. El lenguaje de programación C++. Addison Wesley, 1998.

Abundiz, agosto 2009. http://programacion1abundiz.blogspot.com/2009/09/generalidades-y-caracteristicas-c_08.html