Tipo de artículo: Artículo original

El análisis inteligente de datos mediante el uso de técnicas de Softcomputing

The intelligent analysis of data through the use of techniques of Softcomputing

Wilber Ortiz Aguilar[1*] https://orcid.org/0000-0002-5860-9041
William Andrés Rodríguez López[2] http://orcid.org/0000-0002-5051-9447
Daniel Douglas Iturburu Salvador[3] http://orcid.org/0000-0002-7198-3986
Elsy Rodríguez Revelo[4] http://orcid.org/0000-0003-4486-0785
Yudenalbis La O Mendoza[5] https://orcid.org/0000-0002-9781-7687

[1] Universidad de Guayaquil, Guayaquil, Ecuador, wilber.ortiza@ug.edu.ec
[2] Universidad de Guayaquil, Guayaquil, Ecuador, willian.rodriguezl@ug.edu.ec
[3]Universidad de Guayaquil, Guayaquil, Ecuador, douglas.iturburus@ug.edu.ec
[4]Universidad de Guayaquil, Guayaquil, Ecuador, elsy.rodriguezr@ug.edu.ec
[5]Instituto Politécnico José Martí, La Habana, Cuba, yudilao75@gmail.com

*** Autor para correspondencia: wilber.ortiza@ug.edu.ec**

Resumen

Softcomputing utiliza la mente humana como modelo a seguir y al mismo tiempo tiene como objetivo formalizar los procesos cognitivos humanos que se utilizan para llevar a cabo diferentes tareas diarias. Para comprender la imperfección de los datos y describir el uso simbiótico de las disciplinas computacionales emergentes, se acuñó el término Softcomputing en los años 90. Las tecnologías bajo este término son tolerantes con la imprecisión, la incertidumbre y la verdad parcial. Bajo la explotación de esta tolerancia subyace la notable capacidad del ser humano para descifrar la letra olvidada, para comprender los matices del lenguaje natural, razón por la cual esta tecnología se utiliza a menudo en el análisis inteligente de datos. El análisis inteligente de datos se define como un proceso complejo de descubrimiento de información o conocimiento útil de los datos. En otras palabras, el análisis inteligente de datos es un proceso no trivial de identificación válida, novedosa y potencialmente útil de los patrones comprensibles en los datos. Este complejo proceso está compuesto por una serie de fases interactivas e iterativas donde el usuario debe tomar decisiones adecuadas a lo largo del proceso. Por ello, en este artículo el objetivo es proponer un conjunto de técnicas enmarcadas dentro del Análisis Inteligente de Datos, con capacidad para tratar datos imperfectos y datos de baja calidad.

Palabras clave: Análisis de datos inteligente, técnicas de Softcomputing, datos imperfectos, datos de baja calidad, toma de decisiones.

Abstract

Softcomputing uses the human mind as a role model and at the same time aims to formalize the human cognitive processes that are used to carry out different daily tasks. To understand the imperfection of the data and describe the symbiotic use of emerging computational disciplines, the term Softcomputing was coined in the 90s. Technologies under this term are tolerant of imprecision, uncertainty and partial truth. Under the exploitation of this tolerance underlies the remarkable ability of humans to decipher the neglected letter, to understand the nuances of natural language, which is why this technology is often used in the intelligent analysis of data. Intelligent Data Analysis is defined as a complex process of discovery of information or useful

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

knowledge from the data. In other words, Intelligent Data Analysis is a non-trivial process of valid, novel and potentially useful identification of the comprehensible patterns in the data. This complex process is composed of a series of interactive and iterative phases where the user must make appropriate decisions throughout the process. For this reason, in this article the objective is to propose a set of techniques framed within the Intelligent Data Analysis, with the ability to deal with imperfect data and data of low quality.

Keywords: *Intelligent Data Analysis, Softcomputing techniques, imperfect data, low quality data, decision making.*

Recibido: 08/02/2021
Aceptado: 25/06/2021

Introducción

El análisis de datos, en todas las áreas del conocimiento, ha sido tradicionalmente un proceso que se realiza de forma manual, por uno o más especialistas familiarizados con los datos y con la ayuda de técnicas estadísticas, que son útiles para obtener resúmenes e informes de un conjunto de datos. Este enfoque ha cambiado como consecuencia del crecimiento del volumen de datos en una multitud de dominios como; librerías digitales, archivos de imágenes, bioinformática, atención médica, finanzas e inversión, fabricación y producción, negocios y marketing, redes de telecomunicaciones, dominios científicos, biometría, etc. Por otro lado, cuando la escala de manipulación, exploración e inferencia de datos va más allá capacidad humana, se necesita la ayuda de tecnologías computacionales para automatizar el proceso.

El avance de las nuevas tecnologías ha permitido el almacenamiento de grandes volúmenes de información compuestos por diferentes tipos de datos, que no siempre son tan precisos y perfectos como sería deseable. La revolución digital ha hecho que la información digitalizada sea más fácil de capturar y más barata de almacenar (Fayyad et al., 1996) , (Mitra et al., 2002). El aumento en el volumen y variedad de información ha dado lugar a la necesidad de una nueva generación de herramientas y técnicas para apoyar la extracción de conocimiento a partir de la información disponible. Actualmente, la disciplina que se encarga de obtener este conocimiento útil a partir de los datos es el Análisis Inteligente de Datos, definido por (Fayyad et al., 1996) como el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos.

La AID es un proceso complejo que incluye no solo la obtención de los modelos que capturan el conocimiento, sino también la evaluación y posible interpretación de los mismos, es de uso frecuente en el procesamiento de datos, debido a que cada vez hay más imperfecciones debido al uso cada vez mayor de cuantificadores vagos o imprecisos (bastante jóvenes, muy antiguos, muy pequeños, demasiado grandes, etc.), se cometen frecuentemente errores en los

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

instrumentos de medición utilizados para obtener dicha información, faltan datos, etc. Si bien la imperfección está presente directamente en los datos, aún es bastante escaso el número de técnicas dentro de la disciplina de la AID que permiten el tratamiento explícito de este tipo de información. De esta forma, si las técnicas no son capaces de lidiar con datos imperfectos, estos datos se transforman en precisos y durante este proceso de transformación es posible que exista una pérdida de información relevante que afecte directamente la calidad de los resultados esperados. Ante el problema de la escasez de técnicas que trabajen directamente con datos imperfectos, dentro de la AID se han desarrollado técnicas que permiten el tratamiento de conjuntos de datos que pueden contener datos explícitamente imperfectos, entre los que destacan el Softcomputing. (Bonissone, 1997).

Con el fin de obtener datos limpios, libres de sonido y consistentes, en la presente investigación se propone la combinación de las metodologías que nos ofrece Softcomputing con las técnicas AID, con el propósito de, por un lado, extender algunas de estas técnicas para sumar la capacidad para trabajar directamente con datos imperfectos y, por otro lado, diseñar nuevas técnicas que sean capaces de trabajar explícitamente con este tipo de datos. Dado que los datos constituyen la materia prima del proceso AID y estos pueden ser imperfectos, se propone incorporar al proceso metodologías tolerantes a la imprecisión, la incertidumbre y la verdad parcial. Estas metodologías se han acuñado con el término Softcomputing.

Con base en lo anterior, el principal objetivo de Softcomputing es aprovechar la tolerancia que implica la imprecisión y la incertidumbre, para lograr manejabilidad, robustez y soluciones de bajo costo. Los principales componentes del Softcomputing son la Lógica Difusa, la Neurocomputación y el Razonamiento Probabilístico, incluyendo este último a los Algoritmos Genéticos, las Redes Bayesianas, los Sistemas Caóticos y algunas partes de la Teoría del Aprendizaje. Los componentes de la lógica difusa, la neurocomputación y el razonamiento probabilístico, la lógica difusa se ocupa principalmente de la imprecisión y el razonamiento aproximado; la Neurocomputación del aprendizaje, y el Razonamiento probabilístico de la incertidumbre y la propagación de probabilidades, (Zadeh, 1994). En otras palabras, el término Softcomputing se refiere a un conjunto de metodologías que tienen como objetivo explotar la tolerancia a la imprecisión y la incertidumbre para lograr soluciones de manejabilidad, robustez y bajo costo. El modelo a seguir por Softcomputing es la mente humana.

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

Materiales y métodos

Softcomputing se ubica como la base teórica del área de Sistemas Inteligentes, y deja claro que la diferencia entre el área de Inteligencia Artificial clásica, y la de Sistemas Inteligentes, es que la primera se apoya en el llamado Hardcomputing, mientras que la segunda sí lo hace en Softcomputing. En el presente trabajo nos enfocamos en las fases de preprocesamiento y minería de datos del Análisis Inteligente de Datos para extender las técnicas existentes y diseñar nuevas técnicas dentro de estas dos fases con el objetivo de que puedan trabajar con datos imperfectos (datos de baja calidad) directamente, sin la necesidad de una transformación previa. La figura 1 muestra el esquema general del trabajo realizado en el campo de la AID sobre datos de baja calidad.

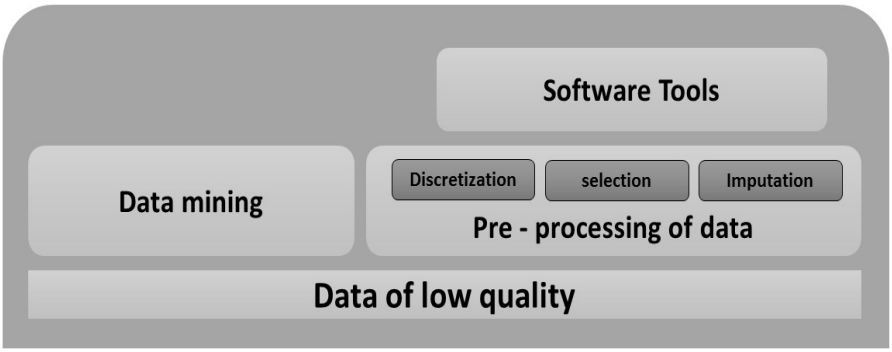

Figura 1. Tareas utilizadas para el tratamiento de datos de baja calidad. Fuente: Self made.

La figura 1, tiene tres fases fundamentales para tratar los datos de baja calidad, estas fases son:

• Fase dirigida a la Minería de datos.

• Fase de preprocesamiento de datos.

• Fase relacionada con las herramientas informáticas desarrolladas para el tratamiento de datos y la obtención de datos de calidad.

La fase dirigida a Data Mining se enfoca en la extensión de un árbol de decisión difuso, ya que los árboles de decisión son una de las opciones más populares para aprender y razonar en base a las características de los ejemplos, (Berzal et al., 2004), (Janikow, 1998), (Myles & Brown, 2003). La popularidad de esta técnica se basa en la facilidad para comprender y analizar los resultados, su eficiencia, la independencia del problema y la capacidad para tratar

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

aplicaciones a gran escala. Los árboles de decisión se incluyen dentro del aprendizaje supervisado, donde los conjuntos de ejemplos están representados por un conjunto de atributos o características (Olaru & Wehenkel, 2003), (Quinlan, 1986), (Umanol et al., 1994).

Algunos de estos ejemplos ya tienen asignada una clase, es decir, ya están clasificados y son ejemplos que se utilizan en la fase de aprendizaje del árbol. El objetivo es obtener una técnica con la que discriminar, describir o taxonomías la tendencia de aquellos ejemplos que no tienen una clase asignada. En el caso mostrado en la figura 1, se propone una extensión del árbol de decisión difusa, pues se basa en lo que necesita una partición difusa con respecto a los atributos numéricos, capaz de trabajar con valores perdidos tanto en atributos nominales como en atributos numéricos. La extensión del árbol de decisión difusa se lleva a cabo para tratar clases inexactas, con valores difusos y de intervalo diferentes a los de la partición difusa inicial, con subconjuntos de valores difusos en atributos numéricos y con subconjuntos de valores difusos y nítidos en atributos nominal. La nueva extensión consiste en la definición de una medida de similitud para calcular el grado de pertenencia de un valor de baja calidad a cada uno de los descendientes de un determinado nodo N. Además, también puede manejar valores difusos en atributos numéricos siempre que estos pertenecen a una partición / discretización de ese atributo.

Algoritmo 1, detalla el proceso de construcción. Si se analizan en detalle algunos aspectos del algoritmo, es necesario mencionar que en el paso 2 a cada ej se le asigna un peso igual a 1 (¬root (ej) = 1) en el nodo raíz, indicando que el ejemplo inicialmente solo se encontró en el nodo raíz del árbol de decisión difuso. Este valor puede continuar siendo 1 siempre que el ejemplo ex no pertenezca a más de un nodo durante el proceso de construcción del árbol. En un árbol de decisión clásico, un ejemplo solo puede pertenecer a un nodo en cualquier momento, por lo que este peso inicial no es necesario, ya que no se modifica a lo largo de la construcción. Por otro lado, para el árbol de decisión difuso, este valor se puede modificar con valores perdidos y con valores numéricos.

Algorithm 1. Learning the Fuzzy Decision Tree

LearningArbolFuzzy (**in:** *E*, *Particiones-F uzzy*; **out:** *Arbol-Decision-F uzzy*)
begin
1.Initialize:*MN* = *A* con los atributos numeee´ricos discretizados de acuerdo a las *P articiones-F uzzy*

1.1 *EN* = *E*

1.2 **for** every example *ej* **do**

$\chi root(ej) = 1$ con $j = 1, \ldots, |E|$
end for
2. *Select an attribute of* ***MN*** *as a test on node* ***N****:*

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

2.1 **for** each attribute *i* **do**

calculate the information gain *GN* , *i* = 1, . . . , |*MN* |
end for

2.2 The attribute is chosen *ibest* whose *GN* is maximum.

2.3 Divide *N* in *Hi* nodes children according to the attribute *ibest*

3. **for** *each node child Nh* with *h* = 1, . . ., *Hi*: **do**

3.1 *MNh* = *MN* − *ibest*

3.2 Get *ENh* de *EN*

3.3 3.3 Go to step 3 if the stop condition on node Nh is not met by doing MN = MNh and EN =

ENh
end for
end

Basado en el algoritmo 1, el árbol se construye cuando se alcanzan las condiciones de parada. Ampliar el aprendizaje y clasificación del árbol de decisión difuso, incorporando nuevos tipos de datos, que son valores de intervalo, valores difusos (o etiquetas) que pueden ser diferentes de los valores difusos que constituyen la partición difusa del atributo y, por tanto, el grado de similitud de estos valores difusos, cada elemento de la partición difusa del atributo puede ser menor que 1 y expresado por subconjuntos nítidos / difusos. Además, permite que el atributo de clase se exprese mediante valores imprecisos a través de un subconjunto nítido / difuso de valores de dominio. La figura 2 muestra estos tipos de valores.

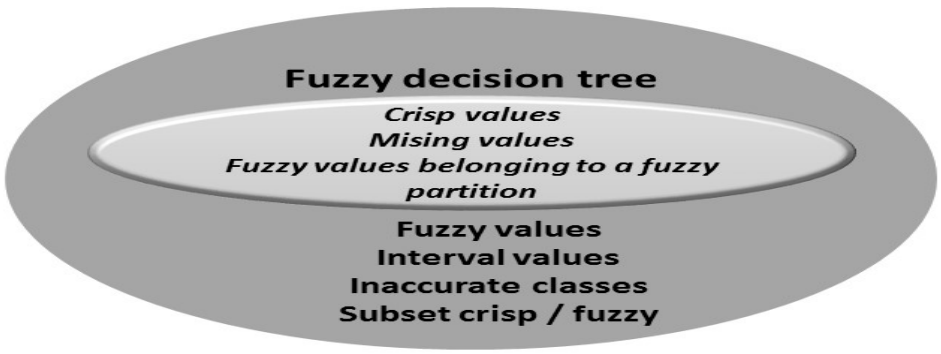

Figura 2. Ampliación de la información procesada por el árbol de decisión difusa. Fuente: Self made

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

Para incorporar nuevos tipos de datos, es necesario modificar los siguientes elementos:

• Definir una función extendida para medir el grado de pertenencia de estos nuevos tipos de datos a las particiones difusas de los atributos numéricos utilizados por el árbol de decisión difusa.

• La definición de esta función requiere la modificación de la función de ganancia de información tanto en la fase de aprendizaje como en la fase de clasificación.

• Para incorporar clases imprecisas, también se deben modificar las fases de aprendizaje y clasificación.

Los resultados obtenidos se evalúan mediante una serie de experimentos. En uno de los experimentos se descubrió que se produce una pérdida de información cuando los datos de baja calidad se transforman en datos nítidos que afectan directamente el rendimiento / precisión de los resultados. En otro de los experimentos se utilizan datos de problemas reales, que contienen datos de baja calidad, y los resultados obtenidos se comparan con otro clasificador de la literatura que también soporta datos de baja calidad, obteniendo resultados satisfactorios y competitivos. Con los experimentos realizados, se puede decir que ambas técnicas son robustas y tienen un comportamiento estable y competitivo frente a otras técnicas presentes en la literatura.

La última técnica extendida es la técnica del vecino k más cercano. Específicamente, la regla de los k-vecinos más cercanos se ha ampliado para poder trabajar con datos de baja calidad. Como elemento principal de esta extensión, se define una medida para calcular la distancia entre ejemplos con valores imperfectos. Más concretamente y que permite trabajar con valores difusos, valores de intervalo, subconjuntos de valores difusos en atributos numéricos y con subconjuntos de valores difusos y nítidos en atributos nominales.

Propuestas detalladas y avances en la fase de minería de datos, nos centramos en la fase de preprocesamiento. En esta fase proponemos un conjunto de técnicas que engloban los procesos de discretización de atributos numéricos, la selección de atributos, la selección de ejemplos y la imputación de valores perdidos.

La fase de preprocesamiento de datos se utiliza con frecuencia cuando se realiza una ampliación de un Fuzzy Random Forest, destacando en esta fase la discretización, selección e imputación de datos. El Fuzzy Random Forest, está compuesto por árboles de decisión difusos para poder trabajar explícitamente con datos de baja calidad, en concreto, trabajamos con los mismos tipos de datos con los que trabaja el árbol de decisión difuso. La discretización de valores numéricos es una etapa crucial en los clasificadores que por su naturaleza no pueden trabajar con datos numéricos, ya que el resultado en la clasificación depende de la calidad de la misma. Además, existen técnicas que, si bien pueden

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

tratar con datos numéricos, obtienen un mejor rendimiento cuando estos valores están discretizados, porque dicha discretización reduce el número de valores numéricos y facilita el aprendizaje más rápido y con mayor precisión. Por estas razones, la técnica de optimización de particiones difusas está diseñada para clasificación, la cual se utiliza para llevar a cabo el proceso de discretización de atributos con valores numéricos mediante la creación de particiones difusas.

La técnica para discretizar atributos numéricos mediante particiones difusas, según (José Manuel Cadenas et al., 2012), (Jose M Cadenas et al., 2012), consiste en una combinación de algoritmos donde se puede taxonomizar la técnica propuesta como supervisada, local, de arriba hacia abajo e incremental, además de utilizar la entropía como medida para obtener y evaluar los intervalos. Esta técnica se compone de dos etapas como se puede observar en la figura 3.

En la primera etapa, se define un conjunto de intervalos precisos mediante la búsqueda de puntos de corte que dividen el dominio. En la segunda, y utilizando los intervalos definidos en la primera etapa, se forman y optimizan las particiones difusas. Para la primera etapa se utiliza un árbol de decisión, mientras que para la segunda se utiliza un algoritmo genético, con el que se determina la cardinalidad y la calidad de las particiones.

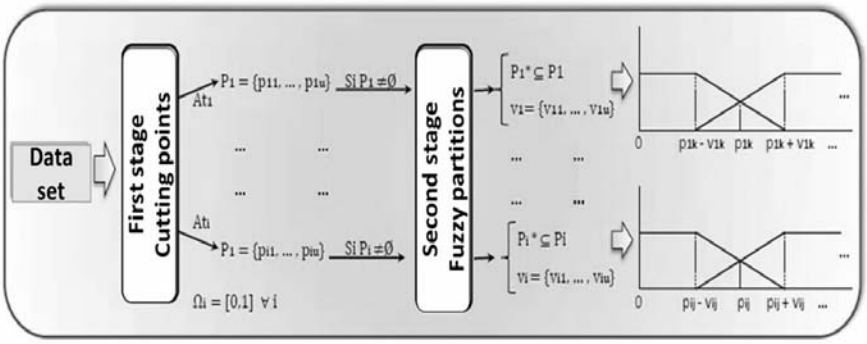

Figura 3. Esquema general del algoritmo para taxonomizar la técnica propuesta como supervisada. Fuente: elaboración propia.

Como se muestra en la Figura 3, en la primera etapa de la técnica de discretización, se construye un árbol de decisión difuso siguiendo las dos modificaciones siguientes como base:

a) La primera es agregar una cola de prioridad en el proceso de aprendizaje del árbol.

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

b) El segundo es que el árbol de decisiones difuso puede tratar con atributos que no están discretizados.

Respecto a la primera modificación, cabe destacar que, al entrar en la cola de prioridad, se examinan los nodos creados en función del peso de los ejemplos que contienen, estudiando primero aquellos con mayor número, ya que contienen mayor cantidad de información.

La segunda modificación, para tratar los atributos numéricos no discretizados, se realiza utilizando el proceso básico del algoritmo C4.5. El umbral seleccionado en cada nodo del árbol de decisión para estos atributos será el punto de corte que delimitará los posibles intervalos para llevar a cabo su discretización. De esta forma, el algoritmo que constituye esta primera etapa se basa en un árbol de decisión difuso que permite trabajar con atributos nominales numéricos discretizados mediante particiones numéricas difusas no discretizadas y además permite la existencia de valores perdidos en todas ellas. El algoritmo 2 describe todo el proceso.

Algorithm 2. *Search for cut points for the technique to discretize numerical attributes through fuzzy partitions.*

Searching Points Cut (in: *E, Partitions-Fuzzy* (optional); out: *Cut-Points*)
begin

1. *Q*=cola vacía, *EN* = *E*, *MN* = *A* (conjunto de atributos), *CP S* vector de tamaño |*A*| inicialmente vacío
2. **for** every example *ej*, *j* = 1, . . . , |*E*|, **do**

   $\chi(ej) = 1$
   end for
3. Create a node N containing all the examples of E with its associated weight. Insert it in Q.
4. **Repeat**

4.1 *N = Get First (Q)*

4.2 **for** each attribute *i* **do**

calculate the information gain GN, i = 1,. . . , | MN |
end for
Choose the ibest attribute whose GN is máximum
if *ibest* is a non-discretized numeric attribute **then**
the selected threshold is a cutoff point for this attribute: *CP Si = CP If J {Cut-Point}*
end if

4.3 Divide N into Hi children nodes according to the attribute *ibest*

4.4 **for** each node child *Nh* con *h* = 1, . . . , *Hi* **do**

Get ENh from *EN*
if (| *ENh* |> mim num examples) and (*ENh* do not have the same class) **then**

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

insert *Nh* en *Q*
end if end for
until (*Q* empty)
5. *Points-Cut =CP S*

end

En esta segunda etapa de la técnica para discretizar atributos numéricos mediante particiones difusas, se utiliza un algoritmo genético para obtener los conjuntos difusos que definen la partición de los atributos numéricos. Los algoritmos genéticos son robustos y tienen mucho poder para lidiar con una gran cantidad de problemas en diferentes áreas, incluida la minería de datos, (Cantú-Paz, 2002), donde no existe una técnica especializada o directamente, aunque la técnica existe, se puede combinar con un algoritmo genético. para mejorar los resultados a través de un algoritmo híbrido, (Cox, 2005) .

En cuanto al proceso de discretización de atributos numéricos, se propone una técnica inicial de discretización y varias mejoras que llevan a cabo la partición de los atributos numéricos mediante particiones difusas. Esta técnica, denominada OFP CLASS, es supervisada, local, de arriba hacia abajo e incremental, además de utilizar la entropía como medida para obtener y evaluar los intervalos. OFP CLASS es una técnica híbrida compuesta por dos etapas:

- En la primera etapa se utiliza un árbol de decisión difuso que obtiene los posibles puntos de corte para dividir el dominio de un atributo
- En la segunda etapa, se utiliza un algoritmo genético para optimizar el número de puntos de corte y construir las particiones difusas finales.

La técnica para discretizar atributos numéricos a través de particiones difusas puede funcionar con atributos nominales y numéricos y ambos atributos pueden contener valores perdidos. Para ampliar los tipos de datos con los que puede trabajar esta técnica, se ha propuesto una mejora de la técnica OFP CLASS, una mejora que permite trabajar directamente con valores de intervalo y valores difusos en los atributos numéricos y también permite el valor de clase de los ejemplos son imprecisos. Otra mejora de la técnica se lleva a cabo con el objetivo de tratar con conjuntos de datos donde el número de ejemplos es proporcionalmente menor que el número de atributos y clases. La parte esencial de esta mejora es la introducción del ensacado tanto en la primera etapa de la técnica como en la segunda etapa. Con la introducción del ensacado, se mejora la calidad de los resultados a medida que se obtiene una mayor riqueza de información en las dos etapas. Tanto la técnica inicial como el resto de mejoras se evalúan mediante una serie de experimentos que muestran la robustez, equilibrio y calidad de los resultados comparados con otras técnicas de la literatura y comparando las mejoras de la técnica entre sí.

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

Centrándonos en el proceso de selección de atributos, hemos propuesto una técnica híbrida para seleccionar atributos a partir de datos de baja calidad. Esto se clasifica como técnica híbrida de filtro - envoltura con búsqueda secuencial hacia adelante en el subconjunto obtenido por la técnica de filtrado y utilizando la clasificación obtenida de estos atributos. Es importante tener en cuenta que, durante todo el proceso de selección de atributos, la técnica no solo proporciona un conjunto óptimo de atributos, sino que en cada etapa proporciona información sobre los atributos que pueden ser relevantes de acuerdo con el objetivo final que se persigue.

En la primera etapa de la técnica, se utiliza un ensamblaje para obtener información sobre la importancia de los atributos. Toda la información recopilada después de usar el ensamblaje se combina utilizando un operador OWA para crear un ranking de importancia de los atributos (ranking que el usuario puede obtener sin necesidad de continuar con la siguiente etapa de la técnica). A partir del ranking obtenido, en la segunda etapa se utiliza un clasificador para obtener el conjunto óptimo de atributos. Es importante señalar que tanto el ensamblaje utilizado en la primera etapa como el clasificador final para la segunda etapa pueden tratar directamente con datos de baja calidad. Los datos de baja calidad con los que se puede trabajar son valores difusos y valores de intervalo en atributos numéricos, valores perdidos tanto en atributos numéricos como nominales, subconjuntos de valores difusos en atributos numéricos y subconjuntos de valores difusos y nítidos en atributos nominales.

Para finalizar la fase de preprocesamiento de los datos, proponemos una técnica de imputación predictiva de valores perdidos basada en la técnica de los k-vecinos más cercanos y una técnica de selección de ejemplos, que son capaces de trabajar explícitamente con datos de baja calidad. La técnica de imputación propuesta es capaz de trabajar con valores de intervalo, con valores difusos, con subconjuntos difusos en atributos numéricos y con subconjuntos difusos y nítidos en atributos nominales. Además, la técnica incluye un parámetro externo donde el usuario expresa el grado de confianza con el que quiere que se lleve a cabo la imputación. Dependiendo del grado de confianza impuesto por el usuario, la imputación de un valor faltante puede llevarse a cabo o no.

Cuando se realiza la imputación de un valor faltante, el valor imputado no tiene que ser un valor nítido, pero puede ser un valor de baja calidad, donde su imperfección dependerá del resto de datos de baja calidad que contengan sus vecinos más cercano. Debido a que la técnica de k vecinos más cercanos tiene un alto costo de procesamiento cuando se trabaja con conjuntos de datos extensos, se desarrolla la técnica de selección de ejemplos, específicamente, denominada técnica de condensación capaz de trabajar con los mismos tipos de datos que la técnica de imputación propuesta, pero tiene la desventaja de que los resultados dependen del orden de los ejemplos en el conjunto de datos.

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

Como medida de orden inicial, la entropía difusa de la clase se utiliza para ordenar los ejemplos de un conjunto de datos en función de la imperfección de su atributo de clase.

Como se muestra en la Figura 1, lo descrito anteriormente se realiza mediante el desarrollo de herramientas de software cuyas herramientas están destinadas a proporcionar un marco común donde se pueda realizar una prueba previa.

Resultados y discusión

Para analizar el rendimiento de la técnica para discretizar atributos numéricos a través de particiones difusas, se seleccionaron 12 conjuntos de datos disponibles en el repositorio de aprendizaje automático de la UCI (machine learning repository, university of california, irvine, school of information and computer sciences)(Asuncion, 2007). Estos conjuntos de datos también se han utilizado en los siguientes trabajos(Choi & Moon, 2007), (Li et al., 2009), (Jordan & Jacobs, 1994). Por tanto, comparamos los resultados obtenidos con la técnica para discretizar atributos numéricos mediante particiones difusas, con los obtenidos mediante las técnicas presentadas. Estas técnicas reflejan diferentes tipos de discretización difusa. Todos los atributos numéricos de estos conjuntos de datos inicialmente no están discretizados y las técnicas deben obtener la discretización para todos ellos. Las técnicas que usaremos son las siguientes:

- ✓ En (Choi & Moon, 2007), se desarrolla un algoritmo genético que optimiza los parámetros para una partición difusa con intervalos adaptativos y también elimina atributos irrelevantes o ruidosos, para reducir la cantidad de datos transformados. Las particiones obtenidas están representadas por la superposición de intervalos. Los valores del atributo se transforman en vectores binarios. Cada vector debe tener como máximo dos, indicando que el valor del atributo del ejemplo pertenece a una o dos particiones, (el hecho de pertenecer a dos elementos de la partición indica la incertidumbre asociada a la misma). Una vez obtenidas las particiones de los atributos, los ejemplos se transforman en vectores de ceros y unos. La función de aptitud es una red neuronal que aprende y clasifica un subconjunto de ejemplos transformados del conjunto de datos.
- ✓ Para la evaluación utilizan dos algoritmos de aprendizaje automático, una red neuronal de retropropagación (ANN) y un árbol de decisión C4.5. Estos algoritmos se denominan FSGAANN y FSGAC4.5 respectivamente. Al igual que el algoritmo de discretización de atributos numéricos mediante particiones difusas (CLASE OFP), esta técnica crea un archivo con las particiones de los atributos continuos, por lo que estas particiones se pueden utilizar a posteriori sin necesidad de ejecutar la técnica cada vez que se necesiten.

Resultados experimentales

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

En (Choi & Moon, 2007), se introduce una técnica para la inducción de árboles de decisión difusos que determina la ubicación y la incertidumbre asociada para cada límite de decisión durante el proceso de construcción del árbol. En este proceso, se genera una partición difusa de los atributos, que proporciona una estimación de confianza de la clasificación a través de la propagación de la función de pertenencia. La superposición de los elementos de la partición está definida por una distribución normal con su media y desviación estándar. Estas distribuciones se obtienen utilizando los mejores puntos de corte de los subconjuntos del conjunto de datos en el nodo. Las particiones de los atributos son específicas del clasificador o del algoritmo que los genera. Estas particiones no pueden ser utilizadas por otros clasificadores.

En (Li et al., 2009), las particiones difusas se construyen mediante la combinación de algoritmos de agrupación difusa mediante la regla del voto de la mayoría. Un algoritmo de emparejamiento de clases basado en los k-vecinos más cercanos proporciona la correspondencia entre las clases de los componentes de la partición difusa. Las particiones difusas resultantes se obtienen a partir de las particiones generadas por cada uno de los algoritmos de algoritmos de agrupamiento difuso, realizando un consenso mediante la regla del voto de mayoría difusa. Las particiones de los atributos son específicas del clasificador o algoritmo que los genera.
En (Li et al., 2009), se construye una técnica similar a la anterior, donde se utilizan los mismos algoritmos. Sin embargo, el método de consenso se lleva a cabo utilizando el voto mayoritario ponderado. En (Asuncion, 2007), se propone FID 3.4. FID 3.4 construye un árbol de decisiones difuso. Además, este documento define el procedimiento para generar particiones a partir de los datos para los atributos numéricos utilizando el mismo árbol de decisión que construye. El procedimiento crea un nuevo archivo con los atributos particionados.

Con estas técnicas y la técnica propuesta para discretizar atributos numéricos a través de particiones difusas (CLASE OFP), se realizaron dos conjuntos de experimentos. En el primero, se compararon los resultados obtenidos por todas las técnicas utilizando las técnicas propuestas por los respectivos autores como evaluador de las particiones. Para evaluar las diferentes particiones obtenidas mediante la discretización de atributos numéricos mediante particiones difusas (CLASE OFP), se utilizó el árbol de decisión difusa, que utiliza las particiones creadas por la técnica para discretizar atributos numéricos mediante particiones difusas (CLASE OFP). El segundo experimento se realiza para comparar las particiones obtenidas por las diferentes técnicas, pero utilizando el mismo evaluador. El evaluador que se utiliza es el árbol de decisiones difuso propuesto por Janikov en (Asuncion, 2007).

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

En general, con los conjuntos de datos utilizados, podemos concluir que las técnicas propuestas son útiles. Cuando un conjunto de datos no verifica la condición, la diferencia fundamental de la nueva técnica de discretización es la partición de los atributos. Los atributos más importantes de la clasificación probablemente estén divididos en más partes y en partes más precisas.

Conclusiones

Las técnicas propuestas y ampliadas han mostrado un comportamiento satisfactorio y optimista tanto cuando se trabaja con datos de baja calidad como cuando se trabaja con datos nítidos. Si bien los resultados obtenidos se pueden catalogar como buenos resultados, aún quedan posibles mejoras por realizar en cada uno de ellos. Lo mismo ocurre con la herramienta de software desarrollada con la que se abre una línea muy interesante en las herramientas disponibles en el campo del Análisis Inteligente de Datos.

Conflictos de intereses

Los autores de la presente contribución declaran que no poseen conflicto de intereses.

Contribución de los autores

1. Conceptualización: Wilber Ortiz Aguilar, William Andrés Rodríguez López.
2. Curación de datos: William Andrés Rodríguez López, Daniel Douglas Iturburu Salvador.
3. Análisis formal: Wilber Ortiz Aguilar, William Andrés Rodríguez López.
4. Adquisición de fondos: Elsy Rodríguez Revelo, Yudenalbis La O Mendoza.
5. Investigación: Daniel Douglas Iturburu Salvador, Elsy Rodríguez Revelo, Yudenalbis La O Mendoza.
6. Metodología: Wilber Ortiz Aguilar, William Andrés Rodríguez López.
7. Administración del proyecto: Wilber Ortiz Aguilar.
8. Recursos: Elsy Rodríguez Revelo, Yudenalbis La O Mendoza.
9. Software: Wilber Ortiz Aguilar, William Andrés Rodríguez López.
10. Supervisión: Yudenalbis La O Mendoza.
11. Validación: Wilber Ortiz Aguilar, William Andrés Rodríguez López, Daniel Douglas Iturburu Salvador.
12. Visualización: Elsy Rodríguez Revelo, Yudenalbis La O Mendoza.

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

13. Redacción – borrador original: Wilber Ortiz Aguilar, William Andrés Rodríguez López, Daniel Douglas Iturburu Salvador, Elsy Rodríguez Revelo, Yudenalbis La O Mendoza.
14. Redacción – revisión y edición: Wilber Ortiz Aguilar, William Andrés Rodríguez López, Daniel Douglas Iturburu Salvador, Elsy Rodríguez Revelo, Yudenalbis La O Mendoza.

Financiamiento

La investigación ha sido financiada por los autores.

Referencias

Asuncion, A. (2007). Uci machine learning repository, university of california, irvine, school of information and computer sciences. *http://www. ics. uci. edu/~ mlearn/MLRepository. html*.

Berzal, F., Cubero, J.-C., Marın, N., & Sánchez, D. (2004). Building multi-way decision trees with numerical attributes. *Information Sciences*, *165*(1-2), 73-90. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.1869&rep=rep1&type=pdf

Bonissone, P. P. (1997). Soft computing: the convergence of emerging reasoning technologies. *Soft Computing*, *1*(1), 6-18. https://sci2s.ugr.es/sites/default/files/files/Teaching/OtherPostGraduateCourses/DataMiningandSoftComputing/Bonissone-Sc.pdf

Cadenas, J. M., Garrido, M. C., & Martínez, R. (2012). Generating optimized fuzzy partitions to classification and considerations to management imprecise data. In *Computational Intelligence* (pp. 151-165). Springer.

Cadenas, J. M., Garrido, M. C., Martínez, R., & Bonissone, P. P. (2012). OFP_CLASS: a hybrid method to generate optimized fuzzy partitions for classification. *Soft Computing*, *16*(4), 667-682.

Cantú-Paz, E. (2002). On the use of evolutionary algorithms in data mining. In *Data Mining: A Heuristic Approach* (pp. 22-46). IGI Global.

Cox, E. (2005). *Fuzzy modeling and genetic algorithms for data mining and exploration*. Elsevier.

Choi, Y.-S., & Moon, B.-R. (2007). Feature selection in genetic fuzzy discretization for the pattern classification problems. *IEICE transactions on information and systems*, *90*(7), 1047-1054.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37-37. https://ojs.aaai.org/index.php/aimagazine/article/download/1230/1131/

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)

Janikow, C. Z. (1998). Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *28*(1), 1-14. http://www.cs.umsl.edu/~janikow/fid/papers/fid.ieeesmc.pdf

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, *6*(2), 181-214. https://www.mitpressjournals.org/doi/pdfplus/10.1162/neco.1994.6.2.181

Li, C., Wang, Y., & Dai, H. (2009). A combination scheme for fuzzy partitions based on fuzzy weighted majority voting rule. 2009 International Conference on Digital Image Processing,

Mitra, S., Pal, S. K., & Mitra, P. (2002). Data mining in soft computing framework: a survey. *IEEE Transactions on Neural Networks*, *13*(1), 3-14. http://library.isical.ac.in:8080/jspui/bitstream/10263/3375/1/Binder1.pdf

Myles, A., & Brown, S. (2003). Induction of decision trees using fuzzy partitions. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *17*(10), 531-536. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.816

Olaru, C., & Wehenkel, L. (2003). A complete fuzzy decision tree technique. *Fuzzy sets and Systems*, *138*(2), 221-254. http://pzs.dstu.dp.ua/DataMining/fuzzy/bibl/A%20complete%20fuzzy%20decision%20tree%20technique.pdf

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106. https://link.springer.com/content/pdf/10.1007/BF00116251.pdf

Umanol, M., Okamoto, H., Hatono, I., Tamura, H., Kawachi, F., Umedzu, S., & Kinoshita, J. (1994). Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, 2113-2118.

Zadeh, L. (1994). Soft computing and fuzzy logic, Software, 11 (6): 48-56.

Esta obra está bajo una licencia *Creative Commons* de tipo Atribución 4.0 Internacional (CC BY 4.0)