TEMat

Este trabajo colaboró con una microcharla durante el XVII *Encuentro Nacional de Estudiantes de Matemáticas*, celebrado en Barcelona en julio de 2016.

Introducción a la lógica difusa y sus aplicaciones

David Lobo Palacios
Universidad de Cádiz
david.lobo@uca.es

Resumen: La lógica clásica es una herramienta insustituible para razonar sobre aspectos del mundo real que son considerados como atemporales, objetivos, estables, etc. Sin embargo, la existencia de afirmaciones que no son totalmente ciertas ni totalmente falsas es evidente. De la imposibilidad de tratar este tipo de afirmaciones con herramientas de la lógica clásica surge la necesidad del desarrollo de las denominadas lógicas no clásicas, entre las cuales se encuentra la lógica difusa.

En este trabajo presentamos una introducción a la lógica difusa, así como algunas aplicaciones de esta lógica en problemas reales: el reconocimiento de textos manuscritos, la investigación espacial en Brasil, la programación lógica y el *big data*.

Palabras clave: lógica difusa, operador lógico, reconocimiento de textos manuscritos, investigación espacial, programación lógica, *big data*.

MSC2010: 03-94.

Recibido: 29 de enero de 2017.
Aceptado: 2 de mayo de 2017.

Referencia: Lobo Palacios, David. «Introducción a la lógica difusa y sus aplicaciones». En: *TEMat*, 1 (2017), págs. 53-67. ISSN: 2530-9633. URL: http://temat.anemat.com/articulo/2017-p53/.

1. Introducción

La principal característica del ser humano que le diferencia del resto de seres vivos es la capacidad de razonar. Dada una serie de premisas, el ser humano es capaz de obtener conclusiones. Por ejemplo, si suponemos cierto «el cielo es azul» y falso «Kepler era dentista», podemos deducir que afirmaciones como «el cielo es azul y Kepler era dentista» o «el cielo no es azul» son falsas. Así mismo, oraciones como «el cielo es azul o Kepler era dentista» o «que el cielo fuera azul no hizo que la población alemana del siglo XVI se hiciera dentista» son verdaderas. En realidad, estamos utilizando operadores lógicos para deducir la veracidad de afirmaciones que han sido construidas a partir de las anteriores. Todos estos operadores se encuentran inmersos en lo que denominamos *lógica clásica*, mediante la cual podemos trabajar con enunciados que son o bien ciertos, o bien falsos.

Sin embargo, ya en la era de Aristóteles se conocía la existencia de afirmaciones que no son totalmente ciertas ni totalmente falsas. La conocida como *paradoja del mentiroso* es un claro ejemplo de ello: «esta oración es falsa». Si suponemos que la oración anterior es verdadera, se deduce que es falsa. Si por el contrario suponemos que es falsa, obtenemos que es también verdadera. En cualquier caso, llegamos a una contradicción. Asimismo, para pronunciarse sobre el valor de verdad de «está lloviendo» hace falta conocer el lugar y el momento en el que se dice esta frase. Otras afirmaciones, como «el chico es listo», tienen una clara componente subjetiva. A un individuo puede parecerle que el chico es listo, pero a Stephen Hawking, no. Además, decir que oraciones como «mañana me tocará la Primitiva» son absolutamente ciertas o absolutamente falsas carece de sentido (por lo general, aunque puede darse el caso de que se jueguen todas las combinaciones posibles).

Con lo cual, cuando se desea representar el conocimiento y diseñar modos de razonamiento sobre aspectos dinámicos del mundo real, hay ocasiones en las que «la verdad de un enunciado no depende solo de la relación entre las palabras del lenguaje y los objetos del mundo, sino también del estado del mundo y el conocimiento de este estado» [22], por lo que se necesita de otras lógicas, las llamadas *lógicas no clásicas*.

La lógica difusa es una extensión de la lógica clásica a contextos en los que encontramos imprecisión o información incompleta. Las bases de la lógica difusa fueron presentadas en 1965 de la mano de Lofti Zadeh, natural de la actual República de Azerbaiyán y profesor de la Universidad de California, en Berkeley, con un artículo titulado «Fuzzy sets» [52]. En este artículo, Zadeh presenta unos conjuntos sin límites precisos, los cuales, según él, juegan un papel importante en el reconocimiento de formas, la interpretación de significados y, especialmente, en la abstracción, la esencia del proceso del razonamiento humano. Posteriormente, otros autores desarrollaron otros aspectos difusos de la matemática, con conceptos como medida difusa [8] o integral difusa [45].

El propósito de este trabajo es proporcionar al lector una noción general sobre la lógica difusa y despertar el interés del mismo en esta materia. Para ello, se mostrarán diferentes problemas reales en los que se han aplicado técnicas difusas. La diversidad de estos problemas permite imaginar el enorme rango de aplicaciones de la lógica difusa, así como la actualidad de dichas aplicaciones.

2. Conjuntos difusos y lógica difusa

La idea propuesta por Lofti Zadeh en 1965 sugiere que la pertenencia a un conjunto es la clave para decidir cuando nos enfrentamos a la incertidumbre. Nótese que la relación entre la lógica y la teoría de conjuntos es muy estrecha. Por ejemplo, en el caso clásico, decimos que «el chico es listo» es verdadero si y solo si el chico pertenece al conjunto de las personas listas.

Específicamente, la propuesta de Zadeh reside en asignar grados de verdad a las afirmaciones. O equivalentemente, asignar grados de pertenencia a un conjunto a cada elemento de un universo. Formalmente:

Definición 1. Sea A un conjunto en un universo U y $f\colon U \to [0,1]$ la función de pertenencia al conjunto A. Si, para cada $x \in U$, $f(x)$ toma valores entre 0 y 1, decimos que A es un **conjunto difuso**. Por el contrario, si, para cada $x \in U$, $f(x)$ solo toma los valores 0 o 1, estamos ante un **conjunto clásico**. ◂

Observación 1. Cuando se define un conjunto clásico, se suele dar una serie de propiedades que han de satisfacer los elementos de un universo para pertenecer al conjunto. Sin embargo, un conjunto difuso se define como una colección de pares de la forma $(x, f(x))$, donde x es un elemento del universo y $f(x)$ es su grado de pertenencia. ◂

Observación 2. En ocasiones se utilizan variantes más generales de la noción de conjunto difuso, considerando valores de pertenencia en una estructura algebraica L diferente al intervalo $[0, 1]$. Usualmente, se requiere que dicha estructura sea un conjunto parcialmente ordenado o un retículo, que no es más que un conjunto parcialmente ordenado tal que cada par de elementos del mismo tiene supremo e ínfimo en dicho conjunto. Este tipo de generalizaciones fueron introducidas por primera vez por Joseph Goguen [23]. ◂

En el siguiente ejemplo se muestra cómo podemos considerar un aspecto cotidiano como la altura de una persona en un ambiente difuso.

Ejemplo 1. Consideremos el universo U de los hombres vivos a 1 de enero de 1970 y sea $h\colon U \to \mathbb{R}^+$ la función que devuelve la altura (en metros) de cada elemento de U. Queremos definir el conjunto A de los hombres altos en U. Proponemos dos opciones para hacerlo:

- Como un conjunto clásico: $A = \{x \in U \mid h(x) \geq 1{,}80\}$.
- Como un conjunto difuso: $A = \{(x, f(x)) \mid x \in U\}$, siendo $f\colon U \to [0, 1]$ la función dada por

$$f(x) = \begin{cases} 0 & \text{si} \quad h(x) \leq 1{,}7, \\ \dfrac{x - 1{,}7}{0{,}1} & \text{si} \quad 1{,}7 < h(x) < 1{,}8, \\ 1 & \text{si} \quad h(x) \geq 1{,}8. \end{cases}$$

Gráficamente, la función de pertenencia del conjunto A quedaría como en la figura 1. Se trata de una función de pertenencia de tipo trapezoidal.

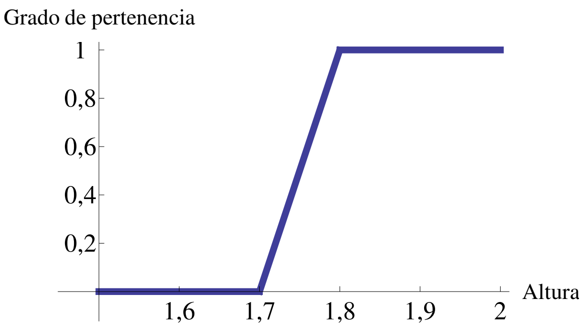

Figura 1: Función de pertenencia trapezoidal.

En el caso clásico, un hombre pertenece al conjunto de los altos si y solo si su altura es mayor o igual a 1,80 m. Ahora bien, un varón que mide 1,79 m podría sentirse molesto por no estar considerado como una persona alta, cuando alguien que mide 1,80 m sí que lo está. En efecto, esto podría ocurrir sea cual sea el listón que pongamos. No parece lógico imponer una altura exacta a partir de la cual decir que alguien es alto.

Cuando consideramos el problema en un ambiente difuso, con la función de pertenencia dada arriba, tenemos que a partir de una altura de 1,80 m una persona es alta (grado de pertenencia 1 a A), al igual que en el caso clásico. De igual manera, tenemos que si la altura de un hombre es menor de 1,70 m, entonces esta persona no es alta (grado de pertenencia 0 a A). Para valores entre 1,70 m y 1,80 m, hemos optado por realizar un ajuste lineal. Así, si alguien mide 1,79 m, entonces tiene un grado de pertenencia 0,9 al conjunto de los altos.

Naturalmente, podríamos tomar cualquier otro ajuste para definir la función de pertenencia. La elección de un ajuste u otro dependerá de la realidad que se desea modelizar. En la figura 2 vemos otra opción para la función de pertenencia, en este caso de tipo gamma.

En cualquier caso, la versión difusa de la definición del conjunto A se ajusta más a nuestra forma de pensar que la versión clásica. La razón principal es que el concepto «altitud» no tiene una definición precisa, tiene un perfil difuso. ◂

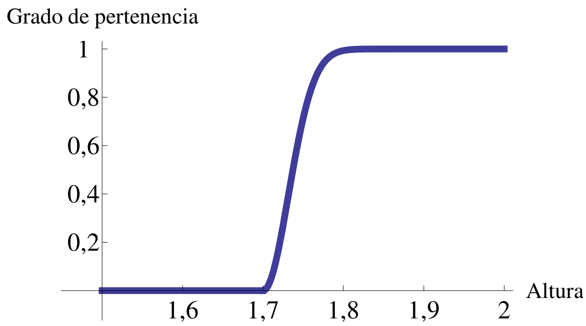

Figura 2: Función de pertenencia gamma.

El objetivo de la lógica difusa es proporcionar un marco de trabajo para representar el conocimiento y obtener reglas de inferencia en un ambiente de incertidumbre e imprecisión léxica. Es decir, queremos no solo asignar valores de verdad a las afirmaciones, sino también trabajar con ellas. Es por esto que necesitamos generalizar los operadores lógicos clásicos. Ahora bien, ¿cómo definimos un operador lógico difuso? Como veremos a continuación, existen infinitas maneras de hacerlo. De nuevo, en función de la realidad que queremos estudiar, tomaremos un operador difuso u otro.

2.1. T-normas y t-conormas

Los operadores lógicos clásicos, aquellos que hemos utilizado (a veces inconscientemente) durante toda nuestra vida, se pueden resumir en conjunción, disyunción, implicación y negación.

En las siguientes líneas, nos centraremos en cómo generalizar la conjunción clásica al caso difuso. Este operador es utilizado a diario mediante la conjunción copulativa «y» del lenguaje natural. Denotado por $\wedge$, queda resumido en el cuadro 1:

Cuadro 1: Conjunción clásica.

p	q	$p \wedge q$
0	0	0
1	0	0
0	1	0
1	1	1

Es decir, $p \wedge q$ (p y q) es cierto si y solo si tanto p como q son ciertos.

Nuestro propósito es definir un operador de $[0, 1] \times [0, 1]$ en $[0, 1]$ que extienda la conjunción clásica. En consecuencia, cualquier conjunción difusa debería satisfacer, al menos, la tabla anterior. Ahora bien, existen infinidad de operadores binarios que cumplen dicha tabla. Algunos ejemplos son los siguientes:

- $p \wedge q = p * q$
- $p \wedge q = \text{mín}(p, q)$
- $p \wedge q = \text{máx}(0, p + q - 1)$

Tomaremos aquella conjunción que mejor se adapte al contexto que queremos estudiar. En ocasiones, exigiremos propiedades adicionales a la conjunción, tales como conmutatividad, asociatividad y monotonía.

En efecto, la generalización de conjunción clásica que ha recibido más atención en los últimos años cumple las propiedades mencionadas. Se trata de la norma triangular o t-norma, la cual definimos a continuación.

Definición 2. Una **norma triangular** (o **t-norma**) es una operación binaria T en el intervalo unidad $[0,1]$, es decir, una función $T\colon [0,1]\times[0,1]\to[0,1]$, tal que para todo $x,y,z\in[0,1]$ se satisfacen las siguientes propiedades:

1. $T(x,y)=T(y,x)$ (conmutatividad);
2. $T(x,T(y,z))=T(T(x,y),z)$ (asociatividad);
3. si $y\le z$, entonces $T(x,y)\le T(x,z)$ (monotonía);
4. $T(x,1)=x$ (condición de frontera). ◂

Huelga decir que la t-norma no es la única generalización de la conjunción clásica en el ámbito difuso. Por ejemplo, la semicópula [3] es un operador menos restrictivo que la t-norma y también generaliza la conjunción clásica.

Observemos que los ejemplos de conjunción difusa mencionados anteriormente son, en realidad, t-normas. De hecho, se trata de las t-normas más utilizadas en la bibliografía.

Ejemplo 2. Las t-normas producto, Gödel y Lukasiewicz son los ejemplos más usuales de este tipo de operador. También es interesante la t-norma drástica, por ser la más pequeña de todas.

$$
\begin{aligned}
&T_P(x,y)=x\cdot y && \text{(t-norma producto)}\\
&T_G(x,y)=\text{mín}(x,y) && \text{(t-norma Gödel)}\\
&T_L(x,y)=\text{máx}(0,x+y-1) && \text{(t-norma Lukasiewicz)}\\
&T_D(x,y)=\begin{cases}0 & \text{si } x,y\in[0,1),\\ \text{mín}(x,y) & \text{en otro caso}\end{cases} && \text{(t-norma drástica)}
\end{aligned}
$$
◂

Además, la t-norma Gödel es la mayor de todas. Esto es, cualquier t-norma toma valores entre la t-norma drástica y la t-norma Gödel. Formalmente:

Proposición 1. *Sea T una t-norma. Entonces, para todo $x,y\in[0,1]$, se verifica que*

$$T_D(x,y)\le T(x,y)\le T_G(x,y). \tag{1}$$

De forma similar, las conormas triangulares o t-conormas generalizan la disyunción clásica (denotada por $\vee$), resumida en el cuadro 2:

Cuadro 2: Disyunción clásica.

p	q	$p\vee q$
0	0	0
1	0	1
0	1	1
1	1	1

En la siguiente definición, podemos observar que las t-normas y t-conormas difieren solamente con respecto a las condiciones de frontera.

Definición 3. Una **conorma triangular** (o **t-conorma**) es una operación binaria S en el intervalo unidad $[0,1]$, es decir, una función $S\colon[0,1]\times[0,1]\to[0,1]$, tal que para todo $x,y,z\in[0,1]$ se satisfacen las siguientes propiedades:

1. $S(x,y)=S(y,x)$ (conmutatividad);
2. $S(x,S(y,z))=S(S(x,y),z)$ (asociatividad);
3. si $y\le z$, entonces $S(x,y)\le S(x,z)$ (monotonía);
4. $S(x,0)=x$ (condición de frontera). ◂

Nótese que la única diferencia entre una t-norma y una t-conorma es que el 1 es el elemento neutro de una t-norma, mientras que en el caso de una t-conorma el 0 es el elemento neutro.

Ejemplo 3. Un ejemplo clásico de t-conorma es el operador máximo, conocido usualmente en la literatura como t-conorma Gödel.

$$S_G(x, y) = \text{máx}(x, y) \qquad \text{(t-conorma Gödel)}$$ ◂

El resultado que sigue muestra cómo obtener una t-conorma a partir de una t-norma y viceversa [25].

Proposición 2. *Una función $S\colon [0,1] \times [0,1] \to [0,1]$ es una t-conorma si y solo si existe una t-norma T tal que para todo $x, y \in [0,1]$ se cumple que*

$$S(x, y) = 1 - T(1 - x, 1 - y). \tag{2}$$

Un ejercicio interesante para familiarizarse con los operadores lógicos difusos es hallar las t-conormas asociadas a las t-normas producto, Gödel, Lukasiewicz y drástica. Naturalmente, la t-conorma obtenida a partir de la t-norma Gödel es la definida en el ejemplo 3.

2.2. Implicaciones difusas y negaciones

Para finalizar esta sección, introduciremos las implicaciones difusas y discutiremos el papel que juega la negación en un paradigma de incertidumbre e información incompleta.

Las implicaciones difusas generalizan la implicación de la lógica clásica conservando las propiedades de monotonía. Siguiendo la nomenclatura usada en programación lógica (una rama de la lógica de la cual hablaremos más adelante), escribiremos «$\leftarrow$» en lugar de «$\rightarrow$».

Definición 4. Decimos que un operador $\leftarrow\colon [0,1]\times[0,1] \to [0,1]$ es una **implicación difusa** si para cada $y, y_1, y_2, z, z_1, z_2 \in [0,1]$ se cumplen las siguientes propiedades:

1. si $z_1 \leq z_2$, entonces $z_1 \leftarrow y \leq z_2 \leftarrow y$. Es decir, $\leftarrow$ es creciente en el consecuente (argumento de la izquierda);
2. si $y_1 \leq y_2$, entonces $z \leftarrow y_2 \leq z \leftarrow y_1$. Esto es, $\leftarrow$ es decreciente en el antecedente (argumento de la derecha). ◂

Como se observa, la definición usual de implicación es muy general. Dependiendo del entorno de aplicación se exigen propiedades adicionales, como es el caso de la implicación residuada, introducida por primera vez por Pavelka [38].

En efecto, si nos paramos a pensar en el papel de la implicación en lógica clásica, resulta un elemento esencial en la regla del *modus ponens*, que es el principal método de deducción. A fin de generalizar también esta regla, se define una nueva implicación que está relacionada con el concepto de operador residual.

Definición 5. Dada una t-norma $T\colon [0,1] \times [0,1] \to [0,1]$, se dice que $\leftarrow$ es una **implicación residuada de T** si existe una implicación difusa $\leftarrow\colon [0,1] \times [0,1] \to [0,1]$ tal que

$$T(x, y) \leq z \ \text{ si y solo si } \ x \leq z \leftarrow y \tag{3}$$

para todo $x, y, z \in [0,1]$. Entonces se escribe como $\leftarrow_T$, y al par $(T, \leftarrow_T)$ se le llama **par residuado** o **par adjunto**. La propiedad dada por la ecuación (3) se conoce como **propiedad de adjunción**. ◂

El siguiente resultado se deduce directamente de la propiedad de adjunción.

Proposición 3. *Sea $(T, \leftarrow_T)$ un par adjunto. Entonces, para cada $y, z \in [0,1]$ se tiene que $y \leq z$ si y solo si $z \leftarrow y = 1$.*

A continuación se muestran las implicaciones adjuntas de las t-normas presentadas en el ejemplo 2.

Ejemplo 4. Las implicaciones residuadas correspondientes a las t-normas producto, Gödel y Lukasiewicz se definen, para todo $y, z \in [0, 1]$, como

$$
\begin{aligned}
z \leftarrow_{\mathrm{P}} y &= \begin{cases} 1 & \text{si } y \leq z, \\ z/y & \text{en otro caso;} \end{cases} \\
z \leftarrow_{\mathrm{G}} y &= \begin{cases} 1 & \text{si } y \leq z, \\ z & \text{en otro caso;} \end{cases} \\
z \leftarrow_{\mathrm{L}} y &= \text{mín}(1, 1 - y + z).
\end{aligned}
$$

◂

El caso del operador de negación en un ambiente lógico no clásico es especialmente peculiar. Un pensamiento común entre los iniciados a las lógicas no clásicas es que no es necesario considerar un operador de negación, puesto que en su lugar podría tomarse un mayor número de conjuntos. Por ejemplo, si A es el conjunto clásico de los científicos y x es un elemento del universo, en lugar de decir que $n(x)$ (la negación de x) pertenece a A, podríamos definir el conjunto B de los no científicos y decir que x pertenece a B.

Sin embargo, por suerte o por desgracia, el ser humano no utiliza siempre la misma negación. En el lenguaje natural se utilizan diferentes tipos de negación en función del sentido que se le da a dicho operador. Pensemos, por ejemplo, en la proposición «Dios existe». A día de hoy, parece imposible demostrar que dicha afirmación es cierta o es falsa. En principio, no es posible asignar un valor de verdad a dicho enunciado. Sin embargo, las personas religiosas (en todas sus vertientes) dirían que dicha afirmación es cierta, mientras que las personas ateas dirían que es falsa. ¿Cómo es posible? La razón es que la comunidad religiosa utiliza información explícita, en este caso sus creencias, para pronunciarse sobre el valor de verdad de la afirmación. Cuando no conoce el valor de verdad de una afirmación, tiene en cuenta algunos aspectos externos a dicha afirmación. Por el contrario, podemos suponer que el grupo de ateos niega una proposición cuando no conoce el valor de verdad de la misma.

Nuestro objetivo es desarrollar una teoría lógica que modelice el uso que hace el ser humano de la negación. Por consiguiente, podemos definir diferentes tipos de negación y diferentes operadores de negación difusos, lo cual aporta a las lógicas no clásicas un grado más de flexibilidad. Algunos de los operadores de negación más importantes son los definidos a partir de implicaciones residuadas de una t-norma [9, 19, 42]; las negaciones ordinarias, estudiadas por Trillas, Esteva y Domingo [16-18, 50], y los pares de negaciones débiles, introducidos por Georgescu y Popescu [21].

En general, un operador de negación difuso viene definido como sigue.

Definición 6. Un operador unario sobre el intervalo unidad $n\colon [0, 1] \to [0, 1]$ se dice que es una **negación** si satisface las siguientes propiedades:

1. Si $x \leq y$, entonces $n(y) \leq n(x)$.
2. $n(0) = 1$, $n(1) = 0$.

Si para todo $x \in [0, 1]$ se satisface $x \leq n(n(x))$, decimos que el operador de negación n es una **negación ordinaria**, mientras que si se verifica la condición $n(n(x)) \leq x$, se trata de una **negación débil**. Decimos que n es una **negación fuerte** (o **involutiva**) si $x = n(n(x))$, para todo $x \in [0, 1]$. ◂

Ejemplo 5. El operador negación más usado es el que se define como $n(x) = 1 - x$, para todo $x \in [0, 1]$, el cual es una negación fuerte. ◂

No es difícil imaginar que, con la generalidad de la teoría que acabamos de desarrollar, las aplicaciones de esta sean muy numerosas y variadas. En la siguiente sección mostramos algunas aplicaciones interesantes de la lógica difusa en problemas reales.

3. Aplicaciones de la lógica difusa

En nuestro día a día tenemos que trabajar con elementos o ideas que no tienen una definición precisa. Hasta el momento, cuando a la hora de resolver un problema encontramos que puede aparecer cierto error, nuestra atención se centra en controlar dicho error. Es decir, asegurar que el error cometido no exceda de ciertos límites.

La lógica difusa surge como una potente herramienta para trabajar con problemas del perfil mencionado. Además, no solo permite disminuir el error cometido (y acotarlo), sino que considera el problema completo desde una nueva perspectiva más general. De esta forma, se aumenta el nivel de abstracción del problema, lo cual conlleva, por lo general, una mejor modelización del mismo.

3.1. Clasificación y reconocimiento de manuscritos

La primera aplicación que mostraremos fue presentada por el profesor László T. Kóczy [26] en el 7.º *Simposio Europeo sobre Inteligencia Computacional y Matemáticas* (ESCIM 2015), celebrado en Cádiz en 2015.

Hoy día es relativamente sencillo encontrar aplicaciones capaces de reconocer textos impresos prácticamente sin cometer ningún error. Sin embargo, cuando se trata de textos manuscritos, la búsqueda se convierte en toda una odisea. No es difícil imaginar el motivo por el que no existen (de momento) aplicaciones de este tipo que aseguren, y efectivamente tengan, una alta precisión. A saber, el umbral de aceptación del usuario en cuanto a la precisión de un sistema de reconocimiento de manuscritos se sitúa en el 97 % [28].

Lo cierto es que reconocer un texto escrito a mano puede resultar bastante complejo, ya que cada persona tiene su propia caligrafía. Por ejemplo, el par de caracteres «cl» puede confundirse fácilmente (dependiendo de la persona que lo escriba) con la letra «d». Así, en castellano, la palabra «clara» podría ser reconocida erróneamente como la palabra «dara», y viceversa.

Usualmente, para alcanzar el nivel de precisión deseado, los métodos de reconocimiento de manuscritos aplican transformaciones geométricas muy complejas; hasta el punto de que algunos métodos recientes no disponen de la suficiente capacidad de cómputo para dar un resultado en un tiempo razonable. Es decir, podríamos obtener un resultado muy bueno, pero dentro de 800 años, lo cual no resulta útil a efectos prácticos. La lógica difusa aparece como una alternativa a estos métodos, proporcionando una solución aceptable para el reconocimiento de manuscritos.

El punto de partida de este trabajo es la consideración de cada carácter escrito a mano como una señal, es decir, una secuencia de movimientos. Para hacerlo, se toma un sistema de coordenadas cartesianas de dimensión 2 en el plano de la hoja donde se encuentra el texto. Posteriormente, se considera cada carácter como una lista cronológicamente ordenada de pares coordenados. De esta forma, es posible utilizar métodos de clasificación y procesamiento de señales para reconocer caracteres manuscritos.

En particular, en la presentación de Kóczy se mostró una familia de reconocedores unitrazo y multitrazo (también influye la cantidad de trazos con la que se escribe un carácter) denominada *reconocedores basados en el difuso* o *FUBAR*, por sus siglas en inglés [46-48]. Tras optimizar el algoritmo FUBAR multitrazo mediante el uso de métodos estadísticos y metaheurísticos, se alcanzó una precisión del 99,49 % [47]. Este porcentaje de acierto es significativamente superior al obtenido por métodos similares anteriores, como el 86,03 % del algoritmo Graffiti 2 obtenido por Költringer y Grechenig [27], o el 96,7 % de la mejor versión del algoritmo $*N* [1].

Una explicación más exhaustiva de todo lo expuesto anteriormente puede encontrarse en el artículo de Tormási y Kóczy [49].

3.2. Investigación espacial en Brasil

La siguiente aplicación de la lógica difusa fue también presentada en el congreso ESCIM 2015, esta vez por parte de Sandra Sandri, investigadora del *Instituto Nacional de Pesquisas Espaciais* (INPE) en Brasil.

Brasil es uno de los países más grandes del mundo, y uno de los más importantes desde el punto de vista biológico. Además de contener el 60 % de la superficie de selva amazónica de la Tierra [36], el 46 % de la matriz energética de este país está formada por energías renovables [44]. Se trata del mayor productor en agricultura tropical y es líder mundial en tecnología de biocombustibles.

Por todo ello, estudiar los efectos de fenómenos como la meteorología, la deforestación y el cambio climático en el territorio brasileño resulta de vital importancia. Estas son algunas de las tareas de las que se encarga el Instituto Nacional de Pesquisas Espaciais, el centro líder en innovación espacial en Brasil. Para tal propósito, el INPE recoge datos de satélites y de diversos sistemas terrestres y, mediante un proceso de investigación y desarrollo tecnológico, proporciona resultados o productos beneficiosos para la sociedad. Cabe destacar su política de acceso libre a los datos, ya que tanto las imágenes como los productos tecnológicos obtenidos por el instituto se encuentran disponibles en internet.

El motor fundamental de los estudios realizados por el INPE es la lógica difusa. A continuación mencionamos algunas aplicaciones recientes de la inteligencia computacional realizadas por el INPE, y en las cuales subyace la lógica difusa.

Para comenzar, destacamos el uso de sistemas neurodifusos para extraer conocimiento de sistemas caóticos. Debido a que numerosos fenómenos del mundo real tienen algún tipo de régimen caótico, estudios como este tienen una gran relevancia. Un sistema neurodifuso [29] no es más que una combinación de dos técnicas: redes neuronales artificiales y sistemas difusos basados en reglas. Mediante herramientas de aprendizaje de redes neuronales se generan reglas de inferencia de un sistema difuso. De esta forma, se maximizan las ventajas de cada técnica y se minimizan sus inconvenientes. Actualmente, estos sistemas están siendo utilizados para generar automáticamente reglas que permitan predecir cambios de régimen en sistemas caóticos, tales como el atractor de Lorenz y el problema de las tres ondas [43]. El objetivo final es aplicar esta metodología para generar reglas de cambio de régimen en el clima de la región nordeste de Brasil.

Por otro lado, se están empleando autómatas celulares difusos en aplicaciones georreferenciadas. La idea de autómata celular fue propuesta por John Von Neumann [37] y consistía en un sistema lógico autorreplicante. En pocas palabras, un autómata celular es un modelo de un sistema dinámico que evoluciona en pasos discretos. El interés de estos modelos es que permite estudiar un gran número de sistemas dinámicos, tales como incendios, inundaciones, fragmentación urbana, deforestación, epidemias o avalanchas.

La principal aportación del INPE en esta línea ha sido la reproducción del círculo SIR (susceptible-infectado-recuperado) para el dengue [12], lo cual puede ser utilizado para comprobar hipótesis sobre controles de calidad.

Antes de concluir este apartado, comentaremos el trabajo desarrollado por el INPE en cuanto al índice de vegetación en el país. En este sentido, el marco de estudio establecido ha sido la aplicación de perceptrones multicapa difusos para la clasificación de series temporales LUCC [39, 40]. Naturalmente, las series temporales que interesan son aquellas relacionadas con el índice de vegetación, como las de las zonas de bosque, de pasto, de cultivo y de deforestación. Otras aplicaciones interesantes de la lógica difusa por parte del INPE han sido relativas a la reparación de satélites y a la obtención de patrones de clorofila del mar.

3.3. Programación lógica

En esta sección introduciremos al lector en lo que podríamos denominar un área de aplicación de la lógica difusa, ya que debido a su extensión se ha convertido en una rama de la lógica difusa en lugar de en una mera aplicación de esta.

El objetivo principal de la programación lógica, como el de tantas otras ramas de la matemática, es entender y modelizar sistemas que cambian en el tiempo, es decir, sistemas dinámicos. A grandes rasgos, un programa lógico podría definirse como un conjunto de reglas, o implicaciones, deducidas de un sistema dinámico. Cada regla contiene el conjunto de premisas que se han de dar para obtener un consecuente [30].

Por ejemplo, supongamos que tenemos un sistema dinámico, digamos una máquina, y deseamos tener en cuenta diversos aspectos de la misma, tales como alta temperatura, comportamiento ruidoso, alto consumo de combustible, etc. Estos aspectos reciben el nombre de *símbolos proposicionales del sistema*. Tras poner en funcionamiento la máquina, observamos que, por ejemplo, cuando la temperatura de la máquina es alta, esta se comporta ruidosamente, y un alto consumo de combustible implica que la temperatura sea alta. Al conjunto de deducciones realizadas lo llamamos programa lógico.

En el caso clásico, consideramos que cada símbolo proposicional solo toma los valores *cierto* o *falso*, y las deducciones que realizamos son totalmente ciertas o totalmente falsas. Sin embargo, como ya hemos visto, en ocasiones los símbolos proposicionales no son ni totalmente ciertos ni totalmente falsos. De la misma forma, podemos tener que la mayoría de las veces que la máquina tiene cierto grado de temperatura elevado, la máquina se comporta ruidosamente (en mayor o menor medida), mientras que otras veces la máquina no hace ningún ruido a pesar de la alta temperatura. Con lo cual, cuando consideramos un programa lógico difuso, el proceso de definición del programa se complica considerablemente, puesto que no solo hemos de deducir reglas de inferencia en el sistema, sino que hemos de dotar de un grado de verdad a dichas reglas [51].

Llegados a este punto, dejaremos de lado la rigurosidad del lenguaje a fin de hacer el texto más adecuado para los no iniciados en la materia. Una introducción formal a la programación lógica difusa requiere la definición de diversas estructuras algebraicas y conceptos que no son absolutamente necesarios para tener una idea general del área, lo cual es el objetivo principal de la presente sección. Aquellos que deseen acceder a más información pueden dirigirse a la bibliografía o contactar con el autor.

El siguiente ejemplo muestra un programa lógico difuso que contiene la información de una máquina similar a la mencionada anteriormente.

Ejemplo 6. Consideremos el retículo $\langle [0,1], \leq \rangle$ y el par adjunto producto $(*, \leftarrow)$.

Sea el conjunto de símbolos proposicionales $\Pi_{\mathbb{P}} = \{p, q, s, t, u\}$ y tomemos el siguiente programa lógico residuado $\mathbb{P}$ (denominado así por tener como operadores un par adjunto, o residuado) valuado en $[0,1]$ y consistente en cinco reglas y un hecho (una regla que no tiene ningún símbolo proposicional en el antecedente):

$$\begin{array}{llll} r_1: \langle p \leftarrow q * \neg s\,; & 0{,}5\rangle, & r_2: \langle q \leftarrow \neg t\,; & 0{,}2\rangle, \\ r_3: \langle t \leftarrow \neg q\,; & 0{,}6\rangle, & r_4: \langle s \leftarrow t * u * \neg p\,; & 0{,}4\rangle, \\ r_5: \langle p \leftarrow q * t\,; & 0{,}6\rangle, & r_6: \langle u \leftarrow\,; & 0{,}4\rangle, \end{array}$$

siendo

$p \equiv$ alto consumo de combustible,
$q \equiv$ bajo nivel de aceite,
$s \equiv$ comportamiento ruidoso,
$t \equiv$ sobrecalentamiento y
$u \equiv$ bajo nivel de agua.

Así, por ejemplo, de la regla r_1 se deduce que en el 50 % de ocasiones en las que el nivel de aceite es bajo y la máquina no se comporta ruidosamente, el consumo de combustible es alto. A partir del resto de reglas se pueden obtener otros comportamientos de la máquina. ◂

Una vez definido un programa lógico, tenemos modelizado el sistema dinámico. El siguiente paso será utilizar dicha modelización para obtener información sobre el sistema. Una herramienta muy interesante para este fin es la técnica del punto fijo.

Nótese que el conjunto de reglas de un programa lógico determina, o, mejor dicho, simula, el comportamiento de un sistema dinámico. Pero el programa es independiente del estado del sistema en un momento determinado, es decir, de los valores de verdad que toman los símbolos proposicionales del sistema en un momento determinado. La pregunta que nos hacemos es «¿podemos encontrar estados del sistema de forma que, aunque el sistema esté en funcionamiento, es decir, que las reglas del programa estén actuando sobre los símbolos proposicionales, siempre tengamos el mismo estado?». Estos estados serán denominados *modelos* del programa lógico.

La respuesta es depende, depende de la estructura algebraica (recordemos la observación 2) y de los operadores lógicos que intervienen en la definición del programa lógico. En particular, con objeto de definir la semántica de un programa lógico, se buscan condiciones bajo las cuales exista un único modelo mínimo, en lugar de un conjunto de modelos minimales. Lo cierto es que, cuando consideramos un programa lógico monótono, es decir, sin ningún operador de negación en el antecedente de las reglas, podemos asegurar la existencia y la unicidad del modelo mínimo [5]. Lo anterior también es cierto para programas lógicos residuados [14] y multiadjuntos [33] (en este caso se consideran varios pares adjuntos en la definición del programa lógico).

Ejemplo 7. El ejemplo 6 es un programa lógico residuado sin negaciones. Por lo tanto, podemos asegurar que tiene un modelo mínimo.

En efecto, realizando los cálculos pertinentes se obtiene que dicho modelo es:

Alto consumo de combustible: 0,0416
Bajo nivel de aceite: 0,0909
Comportamiento ruidoso: 0,0836
Sobrecalentamiento: 0,5454
Bajo nivel de agua: 0,4 ◂

Desafortunadamente, al incluir un operador de negación en la estructura algebraica sobre la que se define un programa lógico, ya sea clásico, difuso o de otro tipo, no podemos asegurar la existencia de modelos del programa (por ejemplo, en el artículo de Madrid y Ojeda-Aciego [31] se prueba lo anterior para el caso residuado). La búsqueda de condiciones que aseguren la existencia y la unicidad de modelos en un entorno de programación lógica con negaciones es un tema de investigación muy activo en la actualidad.

En el caso clásico, Constantini [13] da una condición sintáctica para asegurar la existencia de modelos. En cuanto a la programación lógica no clásica, debido a la pluralidad de teorías no clásicas existente, existen numerosos resultados acerca de la existencia de modelos.

En particular, el estudio de la existencia y unicidad de modelos de un programa lógico residuado con negaciones definido en $[0, 1]$ ha sido realizado previamente por Madrid y Ojeda-Aciego [32]. Recientemente, Cornejo, Lobo y Medina [10] han generalizado los resultados obtenidos en el trabajo anteriormente mencionado. Además, en una continuación de dicho trabajo, se ha desarrollado un entorno de programación multiadjunta con negaciones y se han obtenido condiciones para la existencia de modelos definidos en un programa lógico multiadjunto con negaciones [11].

3.4. *Big data*

La siguiente y última aplicación de la lógica difusa que mostraremos se refiere al *big data*, uno de los marcos de trabajo que más repercusión, tanto científica como mediática, ha tenido en los últimos tiempos.

Hoy en día, seis mil millones de fuentes digitales generan varios zettabytes (10^{21} bytes) de datos al año, y se espera que el número de fuentes aumente a cincuenta mil millones en 2020 [35]. Actualmente se dispone de herramientas para almacenar dicha información, pero desafortunadamente no hay forma de manejarla, y de nada vale guardar información que no se puede utilizar. Ante la incapacidad de manejar tal volumen de datos, se analizan y se extraen patrones ocultos en los datos que nos permiten resumir o deducir la información total contenida en ellos.

Sin embargo, la tecnología requerida para procesar grandes cantidades de datos suele ir acompañada de un alto coste computacional. En consecuencia, se requieren estrategias más eficientes que los métodos tradicionales. Esta es la principal motivación por la que algunas de las mayores compañías del panorama actual, tales como IBM [2], Oracle [41], Microsoft [24] y Google [6, 7, 15, 34], están invirtiendo ingentes cantidades de dinero con el objetivo de optimizar las herramientas de control de flujo de información.

En el mundo del *big data* resulta esencial establecer una jerarquía en los datos de los cuales se desea extraer información. Una estrategia habitual a la hora de elaborar esta clasificación consiste en definir una serie de características, presentes en algunos de los datos, y deducir qué características hacen que un dato pertenezca a una clase o a otra. Actualmente se estudian técnicas de clasificación de datos que consideren dichas características con un perfil difuso [4, 20].

Por ejemplo, supongamos que nuestro objetivo es clasificar un conjunto de textos (en ciertas clases que hemos definido previamente). Las características de los textos en las que nos basaremos para clasificarlos podrían ser palabras clave o grupos de palabras. De esta forma, la aparición de ciertas palabras en un texto nos indica a qué clase pertenece dicho texto.

El proceso de inferencia a partir del conjunto de palabras clave se lleva a cabo mediante algún algoritmo de clasificación, como la máquina de vectores de soporte (SVM) o el clasificador bayesiano ingenuo (*naive-Bayes*). El porcentaje de acierto en la clasificación de los textos depende tanto del algoritmo usado como del conjunto de textos. Así mismo, observemos que una selección adecuada de palabras clave resulta esencial para obtener un buen resultado en la clasificación.

Ahora bien, ¿qué quiere decir que una palabra clave *aparezca* en un texto? Si una palabra clave se encuentra un número elevado de veces en un texto, debería tener más relevancia para su clasificación que otra que solo está presente una vez. Aún más, la aparición de una palabra clave debería tener más peso si lo hace en el título o en el encabezado del texto que si lo hace en el cuerpo del mismo, por ejemplo. Podemos cuantificar todos estos aspectos haciendo uso de técnicas difusas en lugar de los algoritmos clásicos de clasificación. Se trata de una idea que aún está en desarrollo, pero que presenta una interesante línea de trabajo, dada la necesidad y la importancia de manejar grandes cantidades de información en la sociedad actual.

4. Conclusiones

En el lenguaje cotidiano, solemos utilizar conceptos o ideas que no tienen una definición precisa. En este trabajo hemos presentado una lógica no clásica que nos permite escribir y trabajar con esta imprecisión matemáticamente. Se han introducido los principales operadores usados en este paradigma lógico y se ha discutido sobre la importancia de los operadores de negación en un ambiente no clásico. Con el objetivo de evidenciar la versatilidad de la teoría lógica presentada, se han mostrado pinceladas acerca de cuatro aplicaciones de la lógica difusa en problemas reales: la clasificación y reconocimiento de manuscritos, la investigación espacial, la programación lógica y el *big data*.

La lógica difusa es una teoría lógica muy moderna, de apenas medio siglo de vida. Pese a ello, estamos viviendo una revolución a pasos agigantados en la inteligencia computacional. El actual rango de aplicaciones es muy amplio, y se encuentra en constante crecimiento. La razón es bien simple: habitamos un mundo en el que, paradójicamente, lo único exacto son las matemáticas. Todo lo demás es impreciso, difuso.

Aprovechamos estas líneas para agradecer al lector el interés mostrado en el presente trabajo, así como le animamos a profundizar en la temática, para lo cual recomendamos emplear las fuentes mencionadas durante el trabajo.

Referencias

[1] Anthony, Lisa y Wobbrock, Jacob O. «A Lightweight Multistroke Recognizer for User Interface Prototypes». En: *Proceedings of Graphics Interface 2010*. GI '10. Ottawa, Ontario, Canada: Canadian Information Processing Society, 2010, págs. 245-252. isbn: 978-1-56881-712-5. url: http://dl.acm.org/citation.cfm?id=1839214.1839258.

[2] Barranco Fragoso, R. *Qué es Big Data*. https://www.ibm.com/developerworks/ssa/local/ im/que-es-big-data/. (Visitado 18-06-2012).

[3] Bassan, Bruno y Spizzichino, Fabio. «Relations among univariate aging, bivariate aging and dependence for exchangeable lifetimes». En: *Journal of Multivariate Analysis* 93.2 (abr. de 2005), págs. 313-339. https://doi.org/10.1016/j.jmva.2004.04.002.

[4] Bing, L. y Chan, K. C. C. «A Fuzzy Logic Approach for Opinion Mining on Large Scale Twitter Data». En: *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. Dic. de 2014, págs. 652-657. https://doi.org/10.1109/UCC.2014.105.

[5] Blair, H. A.; Brown, A. L. y Subrahmanian, V. S. «Monotone Logic Programming». En: *Intensional Logics for Programming*. Ed. por Cerro, L. Farinas del y Penttonen, M. Oxford: Clarendon Press, 1992, págs. 1-22.

[6] Burrows, Mike. «The Chubby Lock Service for Loosely-coupled Distributed Systems». En: *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*. OSDI '06. Seattle, Washington: USENIX Association, 2006, págs. 335-350. isbn: 1-931971-47-1. url: http://dl.acm.org/citation.cfm?id=1298455.1298487.

[7] CHANG, Fay; DEAN, Jeffrey; GHEMAWAT, Sanjay; HSIEH, Wilson C.; WALLACH, Deborah A.; BURROWS, Mike; CHANDRA, Tushar; FIKES, Andrew y GRUBER, Robert E. «Bigtable: A Distributed Storage System for Structured Data». En: *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*. OSDI '06. Seattle, WA: USENIX Association, 2006, págs. 15-15. URL: http://dl.acm.org/citation.cfm?id=1267308.1267323.

[8] CHOQUET, Gustave. «Theory of capacities». eng. En: *Annales de l'institut Fourier* 5 (1954), págs. 131-295. URL: http://eudml.org/doc/73714.

[9] CINTULA, Petr; KLEMENT, Erich Peter; MESIAR, Radko y NAVARA, Mirko. «Residuated logics based on strict triangular norms with an involutive negation». En: *Mathematical Logic Quarterly* 52.3 (2006), págs. 269-282. ISSN: 1521-3870. https://doi.org/10.1002/malq.200510032.

[10] CORNEJO, M.E.; LOBO, D. y MEDINA, J. «Stable models in normal residuated logic programs». En: *7th European Symposium on Computational Intelligence and Mathematices (ESCIM 2015)*. Ed. por Kacprzyk, J.; Kóczy, László T y Medina, J. Cádiz, Spain, 2015, págs. 150-155.

[11] CORNEJO, M.E.; LOBO, D. y MEDINA, J. «Towards Multi-adjoint Logic Programming with Negations». En: *8th European Symposium on Computational Intelligence and Mathematics (ESCIM 2016)*. Ed. por Kóczy, László T y Medina, J. Sofía, Bulgaria, 2016, págs. 24-29.

[12] COSTA, Walley da; MEDEIROS, Líliam y SANDRI, Sandra. «A Fuzzy Cellular Automata for SIR Compartmental Models». En: *Fuzzy Logic and Applications: 10th International Workshop, WILF 2013, Genoa, Italy, November 19-22, 2013. Proceedings*. Ed. por Masulli, Francesco; Pasi, Gabriella y Yager, Ronald. Cham: Springer International Publishing, 2013, págs. 234-247. ISBN: 978-3-319-03200-9. https://doi.org/10.1007/978-3-319-03200-9_24.

[13] COSTANTINI, S. «On the existence of stable models of non-stratified logic programs». En: *Journal of Theory and Practice of Logic Programming* (2006), 6(1-2):169-212.

[14] DAMÁSIO, Carlos Viegas y PEREIRA, Luís Moniz. «Monotonic and Residuated Logic Programs». En: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 6th European Conference, ECSQARU 2001 Toulouse, France, September 19–21, 2001 Proceedings*. Ed. por Benferhat, Salem y Besnard, Philippe. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, págs. 748-759. ISBN: 978-3-540-44652-1. https://doi.org/10.1007/3-540-44652-4_66.

[15] DEAN, Jeffrey y GHEMAWAT, Sanjay. «MapReduce: Simplified Data Processing on Large Clusters». En: *Commun. ACM* 51.1 (ene. de 2008), págs. 107-113. ISSN: 0001-0782. https://doi.org/10.1145/1327452.1327492.

[16] ESTEVA, F. «Negaciones en retículos completos». En: *Stochastica* I (1975), págs. 49-66.

[17] ESTEVA, F. y DOMINGO, X. «Sobre funciones de negación en [0,1]». En: *Stochastica* IV (1980), págs. 141-166.

[18] ESTEVA, F.; TRILLAS, E. y DOMINGO, X. «Weak and Strong Negation Functions in Fuzzy Set Theory». En: *Proc. XI Int. Symposium on Multivalued Logic*. 1981, págs. 23-26.

[19] ESTEVA, Francesc; GODO, Lluis; HÁJEK, Petr y NAVARA, Mirko. «Residuated fuzzy logics with an involutive negation». English. En: *Archive for Mathematical Logic* 39.2 (2000), págs. 103-124. ISSN: 0933-5846. https://doi.org/10.1007/s001530050006.

[20] FERNÁNDEZ, A.; CARMONA, C.J.; JESUS, M.J. del y HERRERA, F. «A View on Fuzzy Systems for Big Data: Progress and Opportunities». En: *International Journal of Computational Intelligence Systems* 9.1 (2016). TIN2014-57251-P, P11-TIC-7765, UJA2014/06/15, págs. 69-80.

[21] GEORGESCU, G. y POPESCU, A. «Non-commutative fuzzy structures and pairs of weak negations». En: *Fuzzy Sets and Systems* 143 (2004), págs. 129-155.

[22] GOCHET, Paul; GRÉGOIRE, Eric; GRIBOMONT, Pascal; LOUIS, Georges; SANCHEZ, Eduardo; SNYERS, Dominique y WODON, Pierre. *From Standard Logic to Logic Programming: Introducing a Logic Based Approach to Artificial Intelligence*. Ed. por Thayse, André. New York, NY, USA: John Wiley & Sons, Inc., 1988. ISBN: 0-471-91838-5.

[23] GOGUEN, J.A. «L-fuzzy sets». En: *Journal of Mathematical Analysis and Applications* 18.1 (1967), págs. 145-174. ISSN: 0022-247X. https://doi.org/http://dx.doi.org/10.1016/0022-247X(67)90189-8.

[24] Granzen, Achim. «CIO considerations for Big Data». En: Microsoft Services, 2012.

[25] Klement; Mesiar y Pap. *Triangular Norms*. Trends in Logic. Kluwer Academic Publishers, 2000.

[26] Kóczy, László T. *Classification and recognition of movement sequences*. 7th European Symposium on Computational Intelligence and Mathematics, Cádiz. 7 de oct. de 2015. URL: http://escim2015.uca.es/program/keynote-speakers/.

[27] Költringer, Thomas y Grechenig, Thomas. «Comparing the Immediate Usability of Graffiti 2 and Virtual Keyboard». En: *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '04. Vienna, Austria: ACM, 2004, págs. 1175-1178. ISBN: 1-58113-703-6. https://doi.org/10.1145/985921.986017.

[28] LaLomia, Mary. «User Acceptance of Handwritten Recognition Accuracy». En: *Conference Companion on Human Factors in Computing Systems*. CHI '94. Boston, Massachusetts, USA: ACM, 1994, págs. 107-108. ISBN: 0-89791-651-4. https://doi.org/10.1145/259963.260086.

[29] Lin, Chin-Teng y Lee, C. S. George. *Neural Fuzzy Systems: A Neuro-fuzzy Synergism to Intelligent Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN: 0-13-235169-2.

[30] Lloyd, J. W. *Foundations of Logic Programming*. New York, NY, USA: Springer-Verlag New York, Inc., 1984. ISBN: 0-387-13299-6.

[31] Madrid, N. y Ojeda-Aciego, M. «On coherence and consistence in fuzzy answer set semantics for residuated logic programs». En: *Lect. Notes in Computer Science* (2009), 5571:60-67.

[32] Madrid, N. y Ojeda-Aciego, M. «On the existence and unicity of stable models in normal residuated logic programs». En: *International Journal of Computer Mathematics* 89.3 (2012), págs. 310-324.

[33] Medina, J.; Ojeda-Aciego, M. y Vojtáš, P. «Multi-adjoint logic programming with continuous semantics». En: *Logic Programming and Non-Monotonic Reasoning, LPNMR'01*. Lecture Notes in Artificial Intelligence 2173. 2001, págs. 351-364.

[34] Melnik, Sergey; Gubarev, Andrey; Long, Jing Jing; Romer, Geoffrey; Shivakumar, Shiva; Tolton, Matt y Vassilakis, Theo. «Dremel: Interactive Analysis of Web-scale Datasets». En: *Proc. VLDB Endow.* 3.1-2 (sep. de 2010), págs. 330-339. ISSN: 2150-8097. https://doi.org/10.14778/1920841.1920886.

[35] Morabito, V. *Big Data and Analytics: Strategic and Organizational Impacts*. Springer International Publishing, 2015. ISBN: 9783319106656. URL: https://books.google.es/books?id=9lx0BgAAQBAJ.

[36] Mundo, Redacción BBC. *Amazonía: la deforestación amenaza la mitad de las especies de árboles, tipo @ONLINE*. 2015. URL: http://www.bbc.com/mundo/noticias/2015/11/151121_amazonia_arboles_extincion_am.

[37] Neumann, John Von. *Theory of Self-Reproducing Automata*. Ed. por Burks, Arthur W. Champaign, IL, USA: University of Illinois Press, 1966.

[38] Pavelka, J. «On fuzzy logic I, II, III». En: *Zeitschr. f. Math. Logik und Grundl. der Math.* 25 (1979).

[39] Pimentel, Toni; Ramos, Fernando M. y Sandri, Sandra. «Using Fuzzy Multilayer Perceptrons for the Classification of Time Series». En: *Fuzzy Logic and Applications: 10th International Workshop, WILF 2013, Genoa, Italy, November 19-22, 2013. Proceedings*. Ed. por Masulli, Francesco; Pasi, Gabriella y Yager, Ronald. Cham: Springer International Publishing, 2013, págs. 60-67. ISBN: 978-3-319-03200-9. https://doi.org/10.1007/978-3-319-03200-9_7.

[40] Pimentel, Toni; Ramos, Fernando M. y Sandri, Sandra. «Classification of Land Use and Land Cover in the Brazilian Amazon Using Fuzzy Multilayer Perceptrons». En: *Int. J. Nat. Comput. Res.* 5.1 (ene. de 2015), págs. 57-71. ISSN: 1947-928X. https://doi.org/10.4018/ijncr.2015010104.

[41] Plunkett, Tom; Macdonald, Brian; Nelson, Bruce; Hornick, Mark; Sun, Helen; Mohiuddin, Khader; Harding, Debra; Mishra, Gokula; Stackowiak, Robert; Laker, Keith y Segleau, David. *Oracle Big Data Handbook*. 1st. McGraw-Hill Osborne Media, 2013. ISBN: 0071827269, 9780071827263.

[42] San-Min, Wang. «Logics for residuated pseudo-uninorms and their residua». En: *Fuzzy Sets and Systems* 218.0 (2013). Theme: Logic and Algebra, págs. 24-31. ISSN: 0165-0114. https://doi.org/http://dx.doi.org/10.1016/j.fss.2012.11.018.

[43] Santos, Pettras Leonardo Bueno dos; Campos Velho, Haroldo Fraga de; Cintra, Rosangela y Sandri, Sandra. «Chaotic Systems Predictability Using Neuro-Fuzzy Systems and Neural Networks with Bred Vectors». En: *Recent Developments and New Directions in Soft Computing*. Ed. por Zadeh, Lotfi A.; Abbasov, Ali M.; Yager, Ronald R.; Shahbazova, Shahnaz N. y Reformat, Marek Z. Cham: Springer International Publishing, 2014, págs. 295-313. isbn: 978-3-319-06323-2. https://doi.org/10.1007/978-3-319-06323-2_19.

[44] Santos-Varela, A. *El mercado de las energías renovables en Brasil, tipo @ONLINE*. 2013. url: http://espanha-brasil.org/img/documentos/16_document.pdf (visitado 01-10-2013).

[45] Sugeno, M. *Theory of fuzzy integrals and its applications*. Tesis doct. Tokyo Institute of Technology, 1974.

[46] Tormási, A. y Kóczy, L. T. «Fuzzy-based multi-stroke character recognizer». En: *2013 Federated Conference on Computer Science and Information Systems*. Sep. de 2013, págs. 671-674.

[47] Tormási, A. y Kóczy, L. T. «Improving the Accuracy of a Fuzzy-Based Single-Stroke Character Recognizer by Antecedent Weighting». En: *Recent Developments and New Directions in Soft Computing*. Ed. por Zadeh, Lotfi A.; Abbasov, Ali M.; Yager, Ronald R.; Shahbazova, Shahnaz N. y Reformat, Marek Z. Cham: Springer International Publishing, 2014, págs. 165-179. isbn: 978-3-319-06323-2. https://doi.org/10.1007/978-3-319-06323-2_11.

[48] Tormási, Alex y Botzheim, János. «Single-Stroke Character Recognition with Fuzzy Method». En: *New Concepts and Applications in Soft Computing*. Ed. por Balas, Valentina Emilia; Fodor, János y Várkonyi-Kóczy, Annamária R. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, págs. 27-46. isbn: 978-3-642-28959-0. https://doi.org/10.1007/978-3-642-28959-0_2.

[49] Tormási, Alex y Kóczy, László T. «Identification of the initial rule-base of a multi-stroke fuzzy-based character recognition method with meta-heuristic techniques». En: *Czasopismo Techniczne* (2015).

[50] Trillas, E. «Sobre negaciones en la teoría de conjuntos difusos». En: *Stochastica* III (1979), págs. 47-60.

[51] Vojtáš, P. «Fuzzy logic programming». En: *Fuzzy sets and systems* 124.3 (2001), págs. 361-370.

[52] Zadeh, L.A. «Fuzzy sets». En: *Information and Control* 8 (1965), págs. 338-353.