

Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación



Trabajo para optar por el Título de  
Licenciado en Ciencia de la Computación

## **Redes Cognitivas Aproximadas - Difusas**

**Autor**

Carlos Alberto Martínez Mosquera

**Tutores**

Msc. Gonzalo Nápoles Ruiz

Msc. Isel Grau Garcia

**2016**

El que subscribe, Carlos Alberto Martínez Mosquera, hace constar que el trabajo titulado “Redes Cognitivas Aproximadas - Difusas” se realizó en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad en Ciencia de la Computación, autorizando que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

---

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y que el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

---

Firma del Tutor

Msc. Isel Grau Garcia

Msc. Gonzalo Nápoles Ruiz

---

Firma del Jefe de Seminario

Dr. Rafael Bello Pérez

## **AGRADECIMIENTOS**

Un resultado científico es un conglomerado de esfuerzos y este trabajo no es una excepción.

Por eso me gustaría agradecer a:

- Mis tutores, admirables profesionales y amigos.
- Mi mamá por su fe y amor, y al resto de mi familia sin excepción.
- Mi compañera de viaje, Anabel Gonzales.
- Todos mis amigos en general, y a los que fueron.
- Todos los profesores que me impartieron clases durante la carrera y antes, y que aportaron a mi formación y mi sed de aprender.

## **RESUMEN**

Las Redes Cognitivas Aproximadas (RCN) son una alternativa para problemas de toma de decisiones. Este modelo emplea una relación de inseparabilidad “débil” para lidiar con la similitud entre objetos. La selección adecuada del umbral de similitud resulta un aspecto sensible para el algoritmo, siendo este el problema de investigación a abordar. El objetivo de este trabajo es incluir el enfoque difuso en el esquema de RCN para suprimir el umbral de similitud, preservando el desempeño de este modelo en problemas de clasificación. Como resultado se crea un nuevo modelo competitivo con el original, el cual elimina el umbral y es más eficiente.

## **ABSTRACT**

Rough Cognitive Networks (RCN) is a decision model that combines the abstract semantic of the three-way decision model with the neural reasoning mechanism of Fuzzy Cognitive Maps for addressing numerical decision-making problems. This model uses an indiscernibility “weak” relation to deal with the similarity between objects. The algorithm is sensitive to right selection of similarity threshold. The aim of this work is to include the fuzzy approach in the RCN scheme to suppress the similarity threshold, preserving the performance of this model in classification problems. A comparative analysis with the original model and standard classifiers show the competitive of the new proposal, which remove the threshold.

## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
1.1    Objetivo general.....	2
1.2    Objetivos específicos .....	2
1.3    Tareas de investigación.....	2
1.4    Justificación de la investigación .....	3
CAPÍTULO 1.  MARCO TEÓRICO.....	4
1.1    Teoría de los Conjuntos Aproximados .....	4
1.1.1    Utilizando una relación de similaridad.....	5
1.1.2    Hibridación de los Conjuntos Aproximados .....	7
1.2    Modelo de Decisión de Tres Alternativas.....	8
1.3    Mapas Cognitivos Difusos.....	9
1.4    Redes Cognitivas Aproximadas.....	12
1.5    Consideraciones finales del capítulo.....	16
CAPÍTULO 2.  REDES COGNITIVAS APROXIMADAS – DIFUSAS.....	17
2.1    Preliminares .....	17
2.2    Granulación de la información .....	18
2.3    Diseño de la topología de la red.....	20
2.4    Inferencia utilizando FRCN.....	22
2.5    Diseño e implementación de las FRCN sobre la plataforma Weka.....	23
2.5.1    Incorporación de un nuevo clasificador a la plataforma Weka .....	24
2.6    Conclusiones parciales.....	26

CAPÍTULO 3. EVALUACIÓN DEL MODELO.....	27
3.1 Selección de la relación de similaridad, t-norma e implicador difuso.....	27
3.1.1 Selección de un implicador y dos t-normas difusas .....	27
3.1.2 Selección de la función de similaridad.....	30
3.2 Análisis del desempeño del modelo.....	31
3.3 Conclusiones parciales.....	35
CONCLUSIONES.....	36
RECOMENDACIONES .....	37
BIBLIOGRAFÍA.....	38

## LISTA DE FIGURAS

Figura 1. Aproximaciones de un conjunto.....	5
Figura 2. Mapa Cognitivo Difuso.....	10
Figura 3. Actualización del grado de activación de un concepto. ....	11
Figura 4. RCN para un problema de toma de decisiones con dos clases de decisión .....	15
Figura 5. FRCN para un problema clasificación de dos clases de decisión .....	21
Figura 6. Precisión del modelo para cada una de las combinaciones.....	29
Figura 7. Precisión de las funciones de similitud .....	30
Figura 8. Precisión alcanzada por los clasificadores. ....	33
Figura 9. Precisión de las RCN para diferentes umbrales de similitud .....	33
Figura 10. Ranking de clasificadores en la prueba de Friedman.....	34

## LISTA DE TABLAS

Tabla 1. T-normas comunes .....	28
Tabla 2. Implicadores difusos comunes .....	28
Tabla 3. Propiedades de las bases de casos adoptadas en la comparación.....	32
Tabla 4. Resultados de las pruebas de Wilcoxon .....	35

## INTRODUCCIÓN

Las Redes Cognitivas Aproximadas (RCN, por sus siglas en inglés) son una alternativa para problemas de toma de decisiones. Estas estructuras cognitivas combinan la habilidad de los Conjuntos Aproximados para manejar incertidumbre causada por inconsistencia, con el mecanismo de inferencia neuronal de los Mapas Cognitivos Difusos (FCM). Desde el punto de vista conexionista, los RCN son modelos recurrentes representados por grafos dirigidos pesados. En este esquema cada concepto denota una región o clase de decisión del sistema; mientras que los pesos en las conexiones determinan la causalidad entre los nodos. De forma general, la idea consiste en mapear una instancia  $O_i$  a un vector de respuesta  $\vartheta \in [0, 1]^M$ , el cual codifica el grado de aceptación de cada decisión. De esta forma es posible resaltar patrones que no son triviales, y que pueden ser difíciles de encontrar por los decisores.

El primer paso de este modelo consiste en determinar los gránulos información a partir de las regiones positiva, negativa y frontera asociada a cada clase de decisión, para lo cual se emplea una relación de inseparabilidad. En la teoría clásica, la relación de inseparabilidad se define como una relación de equivalencia. Por lo tanto, dos objetos serán inseparables si ellos son idénticos respecto a un conjunto de atributos, pero en algunas situaciones puede resultar demasiado estricto. Este problema es aún más sensible donde los atributos predictores son numéricos; es poco probable que dos atributos tengan el mismo valor decimal. Por lo tanto, dos objetos inseparables pueden ser clasificados como separables.

Para lidiar con este problema se propuso sustituir la relación de equivalencia por una relación de inseparabilidad más “débil” que permita comparar atributos numéricos. La selección adecuada del umbral de esta nueva relación de similaridad resulta un aspecto sensible para el algoritmo, siendo este el problema de investigación a abordar.

### **1.1 Objetivo general**

Incluir el enfoque difuso en el esquema de Redes Cognitivas Aproximadas (RCN) para suprimir el umbral de similaridad, preservando el desempeño de este modelo en problemas de clasificación.

### **1.2 Objetivos específicos**

1. Diseñar un método para eliminar el umbral de similaridad en las RCN utilizando el enfoque difuso.
2. Implementar el modelo propuesto incorporándolo a la plataforma Weka como un nuevo clasificador.
3. Evaluar experimentalmente la propuesta comparando su desempeño contra diferentes clasificadores de la literatura, a través de un conjunto de bases de conocimiento tradicionales para problemas de clasificación.

### **1.3 Tareas de investigación**

1. Realizar una revisión sobre los conceptos relevantes y modelos relacionados con las RCN.
2. Modelar un nuevo algoritmo de aprendizaje a partir de las RCN utilizando un enfoque difuso.
3. Estudiar las clases básicas de Weka, que permiten la creación de un método de clasificación.
4. Implementar las RCN para el enfoque difuso propuesto, basándose en las clases básicas de Weka.
5. Incorporar el nuevo método como un clasificador más de Weka.
6. Diseñar experimentos que permitan la comparación estadística del modelo propuesto contra el modelo original y otros clasificadores de la literatura.

7. Evaluar el desempeño del algoritmo en problemas de clasificación de patrones, usando bases de conocimientos tradicionales.

#### **1.4 Justificación de la investigación**

En el Laboratorio de Inteligencia Artificial de la UCLV se ha trabajado en el desarrollo de sistemas inteligentes y métodos de aprendizaje que hacen uso de técnicas de la llamada computación blanda (*SoftComputing*), logrando el desarrollo de modelos computacionales flexibles, eficientes y eficaces en la resolución de problemas complejos, como lo son las RCN.

Las RCN requieren de un único parámetro: el umbral de similaridad, el cual limita su aplicabilidad. Basados en las RCN, este estudio aborda un nuevo enfoque que resulta de gran utilidad práctica al brindar una mayor flexibilidad al modelo original, pues hacer posible aplicar el método manteniendo su desempeño y sin la necesidad de estimar el umbral de similaridad, un proceso que requiere la intervención de un experto del conocimiento, o bien un procesamiento computacional costoso.

## CAPÍTULO 1. MARCO TEÓRICO

En este capítulo se hace una revisión de varios conceptos y modelos importantes para este estudio, tales como los conjuntos aproximados, el modelo de decisión de tres alternativas, los mapas cognitivos difusos y las redes cognitivas aproximadas.

### 1.1 Teoría de los Conjuntos Aproximados

La Teoría de los Conjuntos Aproximados (RST, por sus siglas en inglés) es una metodología propuesta a principios de los 80's para tratar incertidumbre causada por inconsistencia en los datos. En esta teoría todo concepto es descrito a través de otros dos conjuntos exactos (Pawlak, 1982) (Bello, Falcon, Pedrycz, & Kacprzyk, 2008) (Abraham, Falcon, & Bello, 2009). Sea  $SD = (U, A \cup \{d\})$  un sistema de decisión donde  $U$  denota un conjunto finito no vacío de objetos llamado *universo*, mientras que  $A$  es un conjunto finito no vacío de rasgos o atributos y  $d \notin A$  denota un atributo de decisión. Cualquier subconjunto  $X \subseteq U$  puede ser aproximado por otros dos conjuntos llamados la *aproximación inferior* y *superior* del conjunto  $X$  (Bello & Verdegay, 2012) y se definen por  $B_*X = \{x : [x]_B \subseteq X\}$  y  $B^*X = \{x : [x]_B \cap X \neq \emptyset\}$  respectivamente, donde  $[x]_B$  denota los objetos inseparables asociados a  $x$ , obtenidos a través de una relación de inseparabilidad construida a partir de  $B \subseteq A$ . En este modelo la relación de inseparabilidad (ej. una relación de equivalencia) es reflexiva, transitiva y simétrica. Note que los atributos son discretos, aunque puede ser generalizado para otros dominios (Véase Sección 1.1.1).

La Figura 1 muestra las aproximaciones de un concepto  $X$  en el universo. La aproximación inferior se define por todos los objetos cuya pertenencia al concepto  $X$  es consistente con toda la información, mientras que la aproximación superior se define por los objetos cuya pertenencia solo es coherente con una parte de la información. En otras palabras, los objetos que pertenecen a  $B_*X$  son sin dudas miembros del concepto  $X$ , y los objetos que pertenecen a  $B^*X$  son miembros posibles del concepto  $X$ .

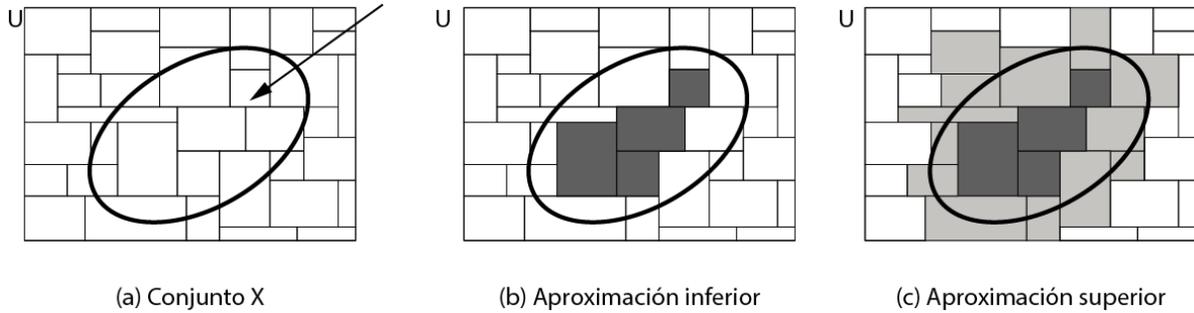


Figura 1. Aproximaciones de un conjunto.

La aproximación inferior  $B_*X$  y superior  $B^*X$  dividen el universo del discurso en tres regiones disjuntas: la aproximación inferior como la *región positiva*, el complemento de la aproximación superior como la *región negativa* y la diferencia entre la aproximación superior e inferior como la *región frontera* (Yao, 2010). La región positiva  $POS(X) = B_*X$  incluye aquellos objetos que son indudablemente miembros de  $X$ , la región negativa  $NEG(X) = U - B^*X$  denota los objetos que no se relacionan con  $X$  y la región frontera  $BND(X) = B^*X - B_*X$  contiene los objetos que su pertenencia a  $X$  es incierta. Estas regiones son en efecto gránulos de información, los cuales son la base del modelo descrito en la Sección 1.4.

La definición de inseparabilidad como una relación de equivalencia es demasiado estricta, de ahí que sea generalizada a otros dominios como se describe a continuación.

### 1.1.1 Utilizando una relación de similitud

En la formulación clásica de los RST (Pawlak, 1982), dos objetos son considerados inseparable si sus atributos tienen valores idénticos. Esta definición funciona bien para atributos nominales, pero es muy estricta cuando los objetos poseen atributos numéricos, por lo que dos objetos pueden ser asumidos incorrectamente como diferentes siendo cercanamente idénticos, afectado aparentemente la consistencia de sistema en investigación. Para contrarrestar esta estricta definición, la relación de inseparabilidad es extendida (Slowinski & Vanderpooten, 2000) y la relación de equivalencia requerida es sustituida por una relación de inseparabilidad más “débil”, que permita comparar atributos numéricos.

Por tanto, dos objetos, de acuerdo a alguna relación de similaridad  $R$ , pertenecerán al mismo conjunto de instancias desiguales, pero razonablemente similares. Esta relación determina cuando dos objetos  $x$  y  $y$  son inseparables o no, y define una clase de similaridad  $\bar{R}(x)$  donde  $\bar{R}(x) = \{y \in U: yRx\}$ . La Ecuación 1.1 muestra la relación de inseparabilidad, donde  $0 \leq \varphi(x, y) \leq 1$  es una relación de similaridad. La relación binaria  $\varphi(x, y)$  considera que  $x$  y  $y$  son inseparables si su grado de similaridad excede el umbral de similaridad  $0 \leq \xi < 1$ . Si  $\xi = 1$  entonces  $R$  conduce al modelo clásico de Pawlak. Si  $\xi < 1$  entonces la relación de similaridad será reflexiva y simétrica, pero no transitiva. La selección adecuada del umbral de similaridad es un aspecto que ha de ser cuidadosamente considerado en el diseño de la relación de similaridad.

$$R: yRx \Leftrightarrow \varphi(x, y) \geq \xi \quad (1.1)$$

La función de similaridad puede ser definida de varias formas. En (Wilson & Martinez, 1997) se estudian numerosas funciones de distancia, las cuales pueden verse como el complemento de funciones de similaridad. En la Sección **Error! Reference source not found.** se analizan varias funciones de distancia. La Ecuación 1.2 formaliza la variante adoptada para este estudio, donde  $A$  es el conjunto de rasgos que describen el problema,  $0 \leq \omega_i \leq 1$  denota la importancia relativa del  $i$ -ésimo atributo,  $x(i)$  y  $y(i)$  denotan el valor del  $i$ -ésimo atributo para el objeto  $x$  y  $y$  respectivamente, y  $\delta(\cdot)$  representa una función para la comparación de atributos. Mientras mayor sea el valor de  $\varphi(x, y)$ , más similares serán los objetos.

$$\varphi(x, y) = \frac{1}{|A|} \sum_{i=1}^{|A|} \omega_i \delta(x(i), y(i)) \quad (1.2)$$

Por otra parte, las Ecuaciones 1.3 y 1.4 muestran las funciones de comparación de rasgos adoptadas. La función  $\delta_1$  es usada para comparar rasgos nominales o discretos, mientras que la  $\delta_2$  es usada para comparar rasgos numéricos, donde  $M_i - m_i$  es el rango del rasgo.

$$\delta_1(x(i), y(i)) = \begin{cases} 1, & x(i) = y(i) \\ 0, & x(i) \neq y(i) \end{cases} \quad (1.3)$$

$$\delta_2(x(i), y(i)) = 1 - \frac{|x(i) - y(i)|}{M_i - m_i} \quad (1.4)$$

En las Ecuaciones 1.5 y 1.6 se definen como obtener las aproximaciones inferior y superior respectivamente, basados en los elementos descritos anteriormente. Nótese que un objeto  $x$  puede pertenecer simultáneamente a más de una clase de similaridad. Por lo tanto, la relación de similaridad  $R$  no conduce a una división del universo  $U$  en particiones o clases de equivalencia, sino genera un cubrimiento sobre el universo en múltiples clases de similaridad  $\bar{R}(x)$  (Bello & Garcia, 2014).

$$B_*X = \{x \in U : \bar{R}(x) \subseteq X\} \quad (1.5)$$

$$B^*X = \bigcup_{x \in X} \bar{R}(x) \quad (1.6)$$

### 1.1.2 Hibridación de los Conjuntos Aproximados

En esta sección se tratará la combinación de los conjuntos aproximados y los conjuntos difusos orientada a problemas de clasificación (Inuiguchi, Wu, Cornelis, & Verbiest, 2013, pp. 4–7) (Jensen & Shen, 2009). Más precisamente, se describirá la generalización difusa de los conjuntos aproximados (FRS, por sus siglas en inglés). En los FRS la relación de equivalencia original es generalizada a una relación de equivalencia difusa. Por tanto, para extender las definiciones de las aproximaciones inferior y superior de una región en el enfoque difuso se define un conjunto difuso  $X \subseteq U$  y una relación binaria difusa  $P \in F(U \times U)$  cuyas funciones de pertenencias están definidas por  $\mu_X(x): U \rightarrow [0, 1]$  y  $\mu_P(y, x): U \times U \rightarrow [0, 1]$ , las cuales representan el grado de pertenencia de  $x \in U$  al concepto  $X$  y en qué medida se presume que  $y$  pertenece al concepto  $X$  dado el hecho que  $x$  es miembro del conjunto difuso  $X$  respectivamente.

Para definir la aproximación inferior se debe considerar el grado de pertenencia de  $x$  al concepto  $X$  y el conocimiento  $P$ . Esto puede ser cuantificado por la certeza de “ $y \in P(x)$  implica que  $y \in X$ ”. En la Ecuación 1.7 se define la función de pertenencia de la aproximación inferior  $P_*(X)$  de  $X$ , donde a función  $I: [0,1] \times [0,1] \rightarrow [0,1]$  es un implicador difuso tal que  $I(0,0) = I(0,1) = I(1,1) = 1$ ,  $I(1,0) = 0$ ,  $I(\cdot, a)$  es decreciente para todo  $a \in [0,1]$  y  $I(\cdot, a)$  es creciente para todo  $a \in [0,1]$ .

$$\mu_{P_*(X)}(x) = \min \left\{ \mu_X(x), \inf_{y \in U} I(\mu_P(y, x), \mu_X(y)) \right\} \quad (1.7)$$

La aproximación superior de la región está dada por la certeza de la sentencia “existe  $y \in X$  tal que  $x \in P(y)$ ”. En la Ecuación 1.8 se define la función de pertenencia de la aproximación superior  $P^*(X)$  de la región, donde  $T: [0,1] \times [0,1] \rightarrow [0,1]$  es una función de conjunción tal que  $T(1,1) = 1$ ,  $T(0,0) = T(0,1) = T(1,0) = 0$  y  $T$  es creciente para ambos parámetros.

$$\mu_{P^*(X)}(x) = \max \left\{ \mu_X(x), \sup_{y \in U} T(\mu_P(x, y), \mu_X(y)) \right\} \quad (1.8)$$

## 1.2 Modelo de Decisión de Tres Alternativas

Yao (Yao, 2010) introdujo el modelo de decisión de tres alternativas al derivar reglas de clasificación de las regiones positivas, negativa y frontera antes mencionadas. Una regla positiva sugiere una decisión de aceptación, una regla negativa significa una decisión de rechazo y una regla de frontera implica una abstención. Más concretamente, sea  $X \subseteq U$  un conjunto a aproximar, la idea de la decisión de tres alternativas es introducida basada en las tres regiones disjuntas producida por las aproximaciones inferior y superior. Basados en estas tres regiones podemos afirmar que cualquier objeto  $x \in POS(X)$  pertenece al concepto  $X$ , y un objetos  $x \in NEG(X)$  no pertenece al concepto  $X$ . Estos principios son la base de las reglas de decisión:

$$Des([x]_B) \rightarrow_P Des(X), si [x]_B \subseteq POS(X)$$

$$Des([x]_B) \rightarrow_B Des(X), si [x]_B \subseteq BND(X)$$

$$Des([x]_B) \rightarrow_N Des(X), si [x]_B \subseteq NEG(X)$$

En estas reglas  $Des([x]_B)$  representa la formula lógica definida por la clase equivalencia  $[x]_B$  y  $Des(X)$  es el concepto. Una regla positiva es utilizada para aceptar, una regla negativa para rechazar, mientras que una regla frontera es una abstención, o sea, si  $[x]_B \subseteq POS(X)$  se acepta al objeto  $x$  como miembro de  $X$ , si  $[x]_B \subseteq NEG(X)$  se rechaza al objeto  $x$  como miembro de  $X$  y si  $[x]_B \subseteq BND(X)$  no se acepta o niega que el objeto  $x$  pertenezca a  $X$  (Yao, 2011). Estas reglas reconocen y modelan la habilidad de hacer una decisión de aceptación o rechazo en escenarios con insuficiente información.

La semántica de las reglas de decisión de tres alternativas es utilizada en la construcción automática de la topología de las RCN, lo cual permite el uso de los elementos de la teoría de los conjuntos aproximados con el razonamiento neuronal en mapas cognitivos difusos en un intento de diseñar un clasificador granular eficiente.

### 1.3 Mapas Cognitivos Difusos

Los Mapas Cognitivos Difusos (FCM, por sus siglas en inglés) son un tipo de red neuronal y fueron introducidos por (Kosko, 1986). Un mapa cognitivo consiste en un conjunto de conceptos (ej. objetos, variables o entidades de algún problema en particular) y sus relaciones causales. El grado de activación de dichos conceptos (o neuronas) normalmente toma valores en el intervalo  $[0, 1]$ . Entre más alto es el grado de activación de una neurona mayor es su influencia sobre el sistema. Por otra parte, la fortaleza de la relación causal entre dos conceptos  $C_i$  y  $C_j$  es cuantificada por un peso numérico  $w_{ij} \in [0,1]$  y es denotado por una arista dirigida desde  $C_i$  a  $C_j$ . Existen tres tipos posibles de relaciones causales que expresan la influencia de una neurona hacia otra, las cuales se muestran a continuación:

- Si  $w_{ij} > 0$ , entonces un incremento (decremento) en el concepto  $C_i$  provocará un incremento (decremento) en el concepto  $C_j$  de forma proporcional al valor  $w_{ij}$ .
- Si  $w_{ij} < 0$ , entonces un incremento (decremento) en el concepto  $C_i$  provocará un decremento (incremento) en el concepto  $C_j$  de forma proporcional al valor  $w_{ij}$ .

- Si  $w_{ij} = 0$ , entonces no existe relación causal entre los dos conceptos.

Desde el punto de vista matemático, un FCM puede ser definido usando una 4-tupla  $(C, W, A, f)$  donde  $C = \{C_1, C_2, C_3, \dots, C_N\}$  representa el conjunto de conceptos del grafo,  $W: (C_i, C_j) \rightarrow w_{ij}$  es una función que asocia un valor causal  $w_{ij}$  a cada par de neuronas  $(C_i, C_j)$ . Este valor  $w_{ij}$  denota la dirección e intensidad de la arista que conecta al concepto  $C_i$  con el concepto  $C_j$ . La matriz de pesos  $W$  define el comportamiento del sistema (Kosko, 1988). De forma similar,  $A: (C_i) \rightarrow A_i$  es una función que asocia un grado de activación  $A_i \in \mathbb{R}$  a cada concepto  $C_i$  del sistema durante el tiempo  $t$  ( $t = 1, 2, \dots, T$ ). Finalmente, este modelo cognitivo usa una función de transformación  $f: \mathbb{R} \rightarrow [0, 1]$  para mantener el grado de activación de cada concepto en el rango adecuado.

El grado de activación de una neurona es un indicador del nivel de presencia del concepto en el sistema modelado (Kosko, 1992). Esta característica es un aspecto clave en la interpretabilidad y usabilidad de los FCM. La Figura 2 muestra la apariencia de un mapa con cinco neuronas, así como sus conexiones.

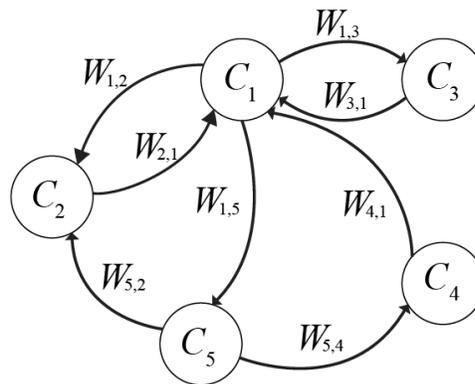


Figura 2. Mapa Cognitivo Difuso

El proceso de propagación de la información (o inferencia) en un FCM se puede definir matemáticamente usando dos componentes: un vector de estado  $A_{1 \times N}$  que representa el grado de activación de los conceptos, y una matriz de pesos causales  $W_{N \times N}$  la cual define la

interacción entre las neuronas. La Ecuación 1.9 define la regla de actualización de los grados de activación, el cual consiste en calcular el vector de estado  $A$  a través del tiempo, para una condición inicial  $A^0$  específica. De forma análoga a otros sistemas neuronales, la activación de  $C_i$  dependerá de la activación las neuronas que inciden directamente sobre el concepto  $C_i$  y de los pesos causales asociados a dicho concepto como se puede observar en la Figura 3.

$$A_i^{(t+1)} = f\left(\sum_{j=1}^N w_{ji}A_j^{(t)}\right), i \neq j \quad (1.9)$$

La función  $f(\cdot)$  en la Ecuación 1.9 denota una función monótona no decreciente y no lineal utilizada para mantener el grado de activación de cada concepto en el rango adecuado. Ejemplos de esas funciones son las variantes de transformación bivalente, trivalente y sigmoideal (Bueno & Salmeron, 2009). En este estudio se utiliza la función sigmoideal  $f(A_i) = 1/(1 + \exp(-\lambda A_i))$  debido a que exhibe una mejor capacidad de predicción (Bueno & Salmeron, 2009).

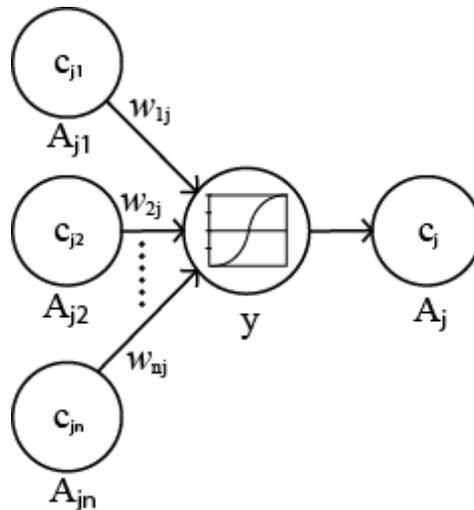


Figura 3. Actualización del grado de activación de un concepto.

Desde el punto de vista algebraico la fase de propagación de un estímulo inicial se reduce a sucesivas multiplicaciones del vector de estado por la matriz de pesos, hasta que se alcance un criterio de parada. Por lo tanto, una variación en un concepto puede afectar directa o indirectamente a todo el mapa (Kosko, 1988). Esto es central en problemas de simulación,

pues permite a los expertos del dominio estudiar el comportamiento del sistema para diferentes condiciones iniciales.

Otra característica que depende del sistema está relacionada con la propagación de la información. Según la Ecuación 1.9 un concepto recibe la influencia de las neuronas conectadas directamente con él, pero no considera el hecho de que el propio concepto puede ser antecedente del grado de activación en cada iteración del proceso de inferencia. La Ecuación 1.10 soluciona este problema, confiriendo al mapa una capacidad memorística adicional. Sin embargo, esta variante en ocasiones conduce a neuronas sobresaturadas, limitando la habilidad para predecir nuevos patrones.

$$A_i^{(t+1)} = f \left( \sum_{j=1}^N w_{ji} A_j^{(t)} + w_{ii} A_i^{(t)} \right), i \neq j \quad (1.10)$$

Dada su naturaleza recurrente, el sistema modelado por un FCM evolucionará durante el tiempo, donde la activación de cada neurona dependerá del grado de activación de sus antecedentes en la iteración anterior. Normalmente este proceso se repite hasta que el sistema estabilice o se alcance un número máximo de iteraciones (Kosko, 1992). Después de un número de iteraciones se arribará a uno de los siguientes estados: (i) estado de estabilidad, (ii) estado cíclico o (iii) estado caótico (Kosko, 1988). En cualquier caso, la salida del FCM corresponde al vector obtenido en la última iteración.

#### 1.4 Redes Cognitivas Aproximadas

Recientemente Nápoles introdujo la teoría de las Redes Cognitivas Aproximadas (RCN, por sus siglas en inglés) (Nápoles, Grau, Falcon, Bello, & Vanhoof, 2015; Nápoles, Grau, Papageorgiou, Bello, & Vanhoof, 2016; Nápoles, Grau, Vanhoof, & Bello, 2014). Las RCN son estructuras cognitivas que combinan la habilidad de los Conjuntos Aproximados para manejar incertidumbre causada por inconsistencia, con el mecanismo de inferencia neuronal de los FCM. En este modelo de gránulos, la semántica de las reglas de decisión de tres alternativas es usada para definir la topología de la red. De forma general, la idea consiste en

mapear una instancia  $O_i$  a un vector de respuesta  $\vartheta \in [0, 1]^M$ , el cual codifica el grado de aceptación de cada decisión.

El primer paso en el proceso de aprendizaje de las RCN es la construcción de los gránulos de información a partir de los datos de entrada usando RST. Las regiones positivas, fronteras y negativas para cada clase de decisión son obtenidas a través de los datos de entrenamiento y una relación de similitud  $R$  predefinida (Véase Sección 1.1.1).

Nótese que cuando se resuelve un problema de clasificación con atributos numéricos la clase de similitud de una nueva instancia  $\bar{R}(x)$  puede activar múltiples reglas de decisión. Si  $\bar{R}(x)$  solo comprende instancias relacionadas a una sola clase de decisión, entonces la clase de decisión estará completamente contenida en la región específica de la decisión. Si la clase de similitud activa la región positiva de múltiples decisiones, las RCN calculan el nivel de confianza para cada una de las reglas y entonces obtiene el grado de preferencia asociado a cada decisión. En los escenarios más complejos, donde solo las regiones fronteras y negativas son activadas, las RCN son capaces de utilizar el conocimiento perteneciente a las regiones positivas, fronteras y negativas para obtener la clase de decisión.

El segundo paso es el diseño automatizado de la topología, donde el FCM sigmoidal es automáticamente construido a partir de los gránulos de información. En este esquema, cada región es mapeada como una neurona de entrada, mientras cada clase de decisión se denota por una neurona de salida. Por lo que, el grafo de RCN contendrá  $3|D| + |D|$  neuronas, donde  $D = \{d_1, d_2, \dots, d_M\}$  es el conjunto de las clases de decisión. Las neuronas de salidas no influyen a otras neuronas debido a que son conceptos receptivos que son utilizados solo para obtener el vector de respuesta. Las reglas siguientes definen la dirección e intensidad de los pesos causales en la topología neuronal propuesta los cuales son estimados a través de las reglas de decisión de tres alternativas:

- (R<sub>1</sub>) Si  $C_i = P_k$  y  $C_j = d_k$  entonces  $w_{ij} = 1.0$
- (R<sub>2</sub>) Si  $C_i = P_k$  y  $C_j = d_{v \neq k}$  entonces  $w_{ij} = -1.0$
- (R<sub>3</sub>) Si  $C_i = P_k$  y  $C_j = P_{v \neq k}$  entonces  $w_{ij} = -1.0$
- (R<sub>4</sub>) Si  $C_i = N_k$  y  $C_j = d_k$  entonces  $w_{ij} = -1.0$
- (R<sub>5</sub>) Si  $C_i = B_k$  y  $C_j = d_v$  y  $BND(X_k) \cap BND(X_v) \neq \emptyset$  entonces  $w_{ij} = 0.5$

En estas reglas  $C_i$  y  $C_j$  denotan dos neuronas,  $P_k$  y  $N_k$  son las regiones positivas y negativas relacionadas a la  $k$ -ésima decisión, mientras que  $-1 \leq w_{ij} \leq 1$  es el peso causal entre la causa  $C_i$  y el efecto  $C_j$ . A través de las regiones fronteras se compone la decisión de abstención.

Concretamente, las reglas R<sub>1</sub> y R<sub>2</sub> definen las relaciones entre las regiones positivas y los conceptos decisión. Si se activa la región  $P_k$  entonces el concepto  $d_k$  se estimula de forma proporcional al estímulo, pues los objetos que pertenecen a la región positiva  $P_k$  serán miembros categóricos del subconjunto  $X_k$ . Siguiendo esa misma lógica, es natural que las decisiones  $d_{v \neq k}$  deben ser inhibidas pues un objeto no puede pertenecer a dos regiones positivas de forma simultánea. La tercera regla R<sub>3</sub> define la interacción entre las regiones positivas: si se activa la región  $P_k$  entonces las demás regiones positivas (aquellas que no están relacionadas con la decisión  $d_k$ ) deben ser inhibidas. Finalmente, si se activa la región negativa  $N_k$  entonces se reducirá el grado de preferencia de la decisión correspondiente, sin embargo, no hay información suficiente para concluir acerca de las demás decisiones involucradas en el problema.

Normalmente las regiones fronteras estimulan la incertidumbre, pero un objeto  $O_i \in BND(X_k)$  podría ser correctamente etiquetado como  $d_k$ . En otras palabras, suponga un problema de toma de decisiones con tres clases y un objeto  $O_i$  sin clasificar, donde  $O_i \in BND(X_1)$ ,  $O_i \in BND(X_2)$  y  $O_i \notin BND(X_3)$ . Esto significa que la instancia podría ser clasificada como  $d_1$  o

$d_2$  de igual forma, pero no hay evidencia alguna que sugiera la clase de decisión  $d_3$ . Basándose en esta observación, la regla  $R_5$  que utiliza esta información en el esquema de decisión de las RCN.

La Figura 4 muestra, como un ejemplo, la RCN que resulta para un problema de toma de decisiones con dos clases de decisión, asumiendo insistencias en el sistema de decisión. Observe que la red utiliza una conexión de “refuerzo” para preservar el conocimiento inicial (el nivel de activación del concepto que se calcula según las características de la entrada) durante la fase de inferencia. Esto significa que la red tiene capacidad memorística que le permite acentuar un patrón inicial.

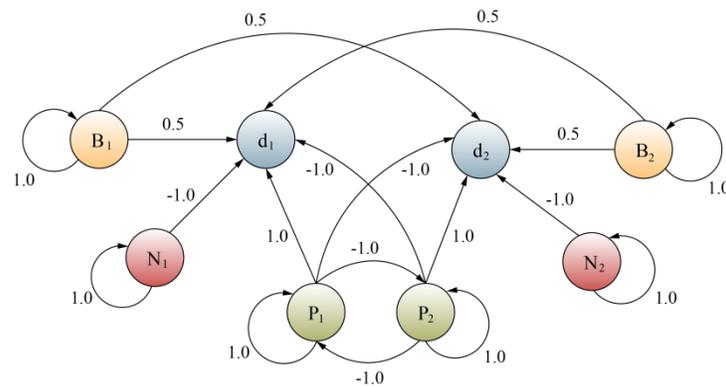


Figura 4. RCN para un problema de toma de decisiones con dos clases de decisión

La principal razón para usar los FCM como mecanismo de inferencia está relacionada con su capacidad para manejar información incompleta o parcialmente contradictoria. Además, los problemas de toma de decisiones usualmente involucran varios conceptos interrelacionados de complejas formas. Esto hace que la retroalimentación del mapa sea fundamental, propagando la información causal a través de rutas no triviales para el decisor. Por otra parte, es conocido que el rendimiento de un FCM depende de los pesos y de la arquitectura definida (Miao & Liu, 2000). Sin embargo, en un RCN el mapa dependerá en última instancia de los datos históricos, y no requiere intervención del experto.

El paso final es relacionado con la explotación de la red. La instancia de entrada  $O_i$  es presentada al clasificador como un vector de entrada  $A^0$  que permite activar la red causal. Las reglas R<sub>6</sub>-R<sub>8</sub> definen el método utilizado para la activación de las neuronas de entrada, el cual se basa en el grado de pertenencia de la nueva instancia a cada región.

$$(R_6) \quad \text{Si } C_i = P_k \text{ entonces } A_i^0 = \frac{|\overline{R}(O_i) \cap POS(X_k)|}{|POS(X_k)|}$$

$$(R_7) \quad \text{Si } C_i = B_k \text{ entonces } A_i^0 = \frac{|\overline{R}(O_i) \cap BND(X_k)|}{|BND(X_k)|}$$

$$(R_8) \quad \text{Si } C_i = N_k \text{ entonces } A_i^0 = \frac{|\overline{R}(O_i) \cap NEG(X_k)|}{|NEG(X_k)|}$$

Una vez obtenido el vector de excitación inicial  $A^0$ , el mecanismo de inferencia descrito a través de la Ecuación 1.9 es aplicado reiteradamente hasta que la red converja a un punto fijo o se alcance un número máximo de iteraciones. Después, la clase con el mayor grado de activación es asignada a la instancia.

### 1.5 Consideraciones finales del capítulo

En este capítulo se realizó una revisión de los conjuntos aproximados y sus extensiones, el modelo de decisión de tres alternativas, los mapas cognitivos difusos y las redes cognitivas aproximadas. Como se ha descrito, las RCN son un modelo granular diseñado especialmente para problemas de clasificación donde las bases de conocimientos manifiesten inconsistencias. Este modelo emplea una relación de inseparabilidad “débil” para lidiar con la similaridad entre objetos, siendo la selección adecuada del umbral de similaridad un aspecto sensible para el algoritmo. Para lidiar con este problema se propone incluir el enfoque difuso en el esquema de RCN para suprimir el umbral de similaridad.

## **CAPÍTULO 2. REDES COGNITIVAS APROXIMADAS – DIFUSAS**

En este capítulo se identifican las principales limitaciones del umbral de similaridad utilizado por las RCN y se aborda una alternativa difusa para lidiar con este problema. Además de describir la implementación de la propuesta para la plataforma Weka.

### **2.1 Preliminares**

Como se ha explicado, las RCN involucran un único parámetro que requiere ser estimado por el experto: el umbral de similaridad. Este parámetro es indispensable para construir las aproximaciones inferiores y superiores. Por lo tanto, el desempeño final de la red dependerá de la correcta estimación de este valor. Si  $\xi$  es muy pequeño, entonces la cardinalidad de las regiones positivas será pequeña. Como resultado las neuronas estarán pobremente excitadas, y por tanto será más complicado determinar la decisión acertada. Si  $\xi$  es demasiado grande, entonces la cardinalidad de las regiones fronteras será también grande, estimulando la incertidumbre durante el proceso de inferencia del FCM Sigmoidal y por tanto reduciendo el rendimiento de la red.

Otra variante de las RCN para el ajuste de parámetros es utilizar un método de aprendizaje basado en la meta-heurística Harmony Search (HS), el cual se enfoca en la búsqueda de parámetros que permitan la construcción de relaciones de similaridad más precisas. Esta meta-heurística es un método de búsqueda de trayectoria simple que evalúa una solución potencial en cada iteración. La evaluación de una solución requiere hallar las aproximaciones inferior y superior de las clases de decisión (para la relación de similaridad estimada), construir la red a partir de los gránulos de información y evaluar la inferencia de la red utilizando validación cruzada. Por lo que este proceso puede llegar a ser computacionalmente costoso.

La adecuada selección del umbral de similaridad resulta un aspecto sensible para el modelo de las RCN y su estimación puede ser computacionalmente costosa, de ahí que se presente en el

siguiente estudio una alternativa difusa para lidiar con este problema, la cual llamaremos Redes Cognitivas Aproximadas-Difusas (FRCN, por sus siglas en inglés).

En esta metodología la primera etapa es la granulación del espacio de ejemplos (ej. el conjunto de objetos de entrenamiento) usando FRS, lo cual comprende la construcción de las regiones para cada clase de decisión considerada en el proceso de clasificación. La segunda etapa está relacionada con el diseño automatizado de la red, donde las neuronas denotan gránulos de información, mientras que las relaciones causales entre esos conceptos son automáticamente estimados. Nótese que este paso no involucra la intervención de un experto, evitando afectaciones negativas por la subjetividad en la construcción del modelo. Finalmente, la tercera etapa describe el proceso de inferencia para explotar la red. Este capítulo describe los cambios realizados al modelo original en la nueva alternativa.

## 2.2 Granulación de la información

Como se ha mencionado, las RCN obtienen las aproximaciones inferior y superior asociadas a cada clase de decisión a través de una relación de inseparabilidad “débil”, la cual requiere de la estimación del umbral de similaridad. Con el objetivo de eliminar este umbral, las FRCN determinan las aproximaciones utilizando los FRS, generalizando la relación de inseparabilidad a una relación de equivalencia difusa.

Similar a las RCN el primer paso de las FRCN está orientado a determinar las regiones positivas, fronteras y negativas. Sea  $X = \{X_1, \dots, X_k, \dots, X_M\}$  una partición del universo  $U$  de acuerdo al valor del atributo de decisión  $d$ , donde cada  $X_k$  es una clase de decisión que contiene todos los objetos clasificados como  $d_k$ . Esta formulación supone un problema de clasificación donde los atributos pueden ser continuos o discretos.

Para determinar las aproximaciones de una clase de decisión  $X_k$ , como se describe en la Sección 1.1.2, es necesario definir la función de pertenencia  $\mu_{X_k}: U \rightarrow [0, 1]$  y la relación

binaria difusa  $\mu_P(y, x): U \times U \rightarrow [0, 1]$ . En las Ecuaciones 2.1 y 2.2 se definen estas funciones respectivamente.

$$\mu_{X_k}(y) = \begin{cases} 1, & y \in X_k \\ 0, & y \notin X_k \end{cases} \quad (2.1)$$

$$\mu_P(y, x) = \mu_{X_k}(x) * \varphi(x, y) \quad (2.2)$$

La Ecuaciones 2.3 y 2.4 definen las aproximaciones inferior y superior de una clase de decisión  $X_k$ , donde  $I(\cdot)$  es un implicador difuso y  $T_1(\cdot)$  es una T-norma difusa. Para este estudio se adoptan los operadores de.

$$\mu_{P_*(X_k)}(x) = \min \left\{ \mu_{X_k}(x), \inf_{y \in U} I(\mu_P(y, x), \mu_{X_k}(y)) \right\} \quad (2.3)$$

$$\mu_{P^*(X_k)}(x) = \max \left\{ \mu_{X_k}(x), \sup_{y \in U} T_1(\mu_P(x, y), \mu_{X_k}(y)) \right\} \quad (2.4)$$

En este estudio, para determinar las aproximaciones, se adoptan los operadores difusos de Lukasiewics, dado que es la mejor combinación de operadores que ajustan el modelo en la Sección 3.1.1. Las Ecuaciones 2.5 y 2.6 definen el implicador y t-norma difusos de Lukasiewics respectivamente.

$$I(x, y) = \min\{1 - x + y, 1\} \quad (2.5)$$

$$T(x, y) = \max\{x + y - 1, 0\} \quad (2.6)$$

Una vez determinadas las aproximaciones, las regiones positiva, negativa y frontera asociadas a clase de decisión  $X_k$  se definen como se muestra en las Ecuaciones 2.7, 2.8 y 2.9 respectivamente.

$$\mu_{POS(X_k)}(x) = \mu_{P_*(X_k)}(x) \quad (2.7)$$

$$\mu_{NEG(X_k)}(x) = 1 - \mu_{P^*(X_k)}(x) \quad (2.8)$$

$$\mu_{BND(X_k)}(x) = \max\{\mu_{P^*(X_k)}(x) - \mu_{P_*(X_k)}(x), 0\} \quad (2.9)$$

Una diferencia significativa de las FRCN respecto al modelo original es que los objetos poseen un determinado grado de pertenencia a cada una las regiones, por lo que las regiones de una clase de decisión no son disjuntas. Posteriormente se mostrará cómo repercute esto en el diseño de la topología de la red.

### 2.3 Diseño de la topología de la red

El segundo paso consiste en el diseño de la topología de la red, al igual que el modelo original las FRCN en esencia es un FCM Sigmoidal donde las neuronas de entrada denotan regiones, mientras que los conceptos de salida representan decisiones o clases del problema investigado. Más precisamente, cada región asociada a un subconjunto  $X_k$  se representa como un nodo de entrada, y son usados para activar la red. Por otra parte, los  $|D|$  neuronas de salida son usados para cuantificar el grado de deseabilidad de cada decisión una vez que se ha propagado el vector de estímulo inicial (este punto será tratado en la próxima Sección 2.4). Estos  $|D|$  conceptos decisión son utilizados para construir el vector de respuesta  $\vartheta \in [0, 1]^M$ , el cual contiene evidencias sobre la decisión final. Una vez que se definen los conceptos del mapa, se establecen conexiones causales entre estos conceptos. La dirección y magnitud de estos valores causales de definen de manera similar al modelo original siguiendo las reglas R<sub>1</sub>-R<sub>4</sub>.

Como se ha mencionado previamente las regiones de una clase de decisión no son disjuntas por lo que la semántica de la decisión de abstención se pierde. Por lo tanto, la regla R<sub>5</sub> es omitida en la nueva alternativa dado que siempre se cumple y no brinda información de utilidad que ayude a discernir entre una clase de decisión u otra en el proceso de inferencia. Al omitirse la R<sub>5</sub> se puede prescindir además de los gránulos de información que representan las

regiones fronteras, debido a que quedan como nodo aislados en el mapa. La Figura 5 muestra el mapa resultante para un problema de toma de decisiones con dos clases de decisión.

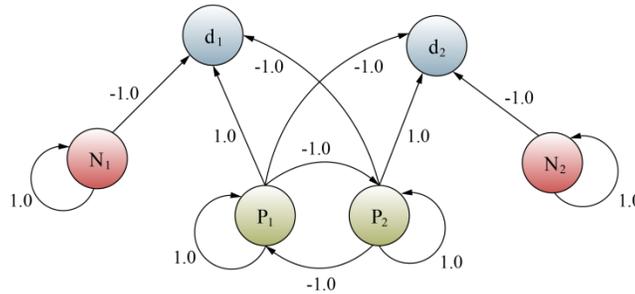


Figura 5. FRCN para un problema clasificación de dos clases de decisión

El Algoritmo 1 describe detalladamente el proceso de construcción de la red para el modelo propuesto.

---

Algoritmo 1. Construcción de la topología de la red

---

- 1 Para toda clase de decisión  $X_k$  hacer
    - 2 Agregar neurona  $P_k$  como la región positiva k-ésima.
    - 3 Agregar neurona  $N_k$  como la región negativa k-ésima.
    - 4 Fin.
  - 5 Para toda decisión  $d_k$  hacer
    - 6 Agregar neurona  $D_k$  como la decisión k-ésima.
    - 7 Fin.
  - 8 Para toda neurona  $C_i$  hacer
    - 9 Para toda neurona  $C_j$  hacer
      - 10 Conectar de acuerdo a las reglas R<sub>1</sub>-R<sub>4</sub>
      - 11 Fin.
    - 12 Fin.
  - 13 Para toda neurona  $C_i \notin D$  hacer
    - 14 Agregar conexión de refuerzo a la neurona  $C_i$  con  $w_{ii} = 1$
    - 15 Fin.
-

## 2.4 Inferencia utilizando FRCN

El último paso del método propuesto es la explotación de la red donde el vector de preferencia  $\vartheta \in [0, 1]^M$  es obtenido, por tanto es necesario calcular el valor de activación de las neuronas de entrada. La instancia  $O_i$  es presentada al clasificador como un vector inicial de entrada  $A^{(0)} = [A(P_1), \dots, A(P_k), \dots, A(P_M), A(N_1), \dots, A(N_k), \dots, A(N_M), A(D_1), \dots, A(D_k), \dots, A(D_M)]$  donde la función  $A(\cdot)$  es utilizada para obtener el grado de excitación inicial de la neurona (Nótese que  $A(D_1) = \dots = A(D_k) = \dots = A(D_M) = 0$ ). Este vector de estímulo  $A^{(0)}$  es calculado a partir de la relación existente entre similaridad la nueva instancia  $O_i$  y las diferentes regiones. El vector  $A^{(0)}$  se obtiene a través de las reglas R<sub>9</sub>-R<sub>10</sub>. Para la agregación de los grados de pertenencia en estas reglas se adopta la t-norma de Lukasiewics definida previamente en la Ecuación 2.6, debido a que ha sido la t-norma que mejor desempeño durante la experimentación en la Sección 3.1.1.

$$(R_9) \quad \text{Si } C_i = P_k \text{ entonces } A_i^0 = \frac{\sum_{x \in U} T_2(\varphi(x, O_i), \mu_{POS}(x_k)(x))}{\sum_{x \in U} \mu_{POS}(x_k)(x)}$$

$$(R_{10}) \quad \text{Si } C_i = N_k \text{ entonces } A_i^0 = \frac{\sum_{x \in U} T_2(\varphi(x, O_i), \mu_{NEG}(x_k)(x))}{\sum_{x \in U} \mu_{NEG}(x_k)(x)}$$

Una vez que se ha estimado el vector de activación, entonces se propaga la información causal usando el mecanismo de inferencia del FCM Sigmoidal hasta que se encuentre un punto fijo, o se alcance un número máximo de iteraciones  $T$ . Este proceso de inferencia acentúa el patrón codificado en la entrada del mapa, lo que resulta relevante en datos inconsistentes donde existen objetos muy similares asociados a decisiones diferentes. A partir de la activación final de las neuronas de salida se construye el vector de respuesta, el cual contiene la preferencia de cada decisión involucrado en el problema. Este vector permite a los expertos ordenar las alternativas según el grado de preferencia que sugiere el modelo. En el caso de un problema de clasificación, la etiqueta del nuevo objeto corresponderá a la clase de decisión con mayor activación. El siguiente algoritmo resume cómo calcular el vector de respuesta a partir del vector de estado inferido por el mapa.

---

**Algoritmo 2. Calcular el vector de preferencia**


---

```

1  Para toda neurona  $C_i$  hacer
2      Calcular el estímulo inicial  $A_i^{(0)}$  de acuerdo a las reglas R9-R10
3  Fin.
4  Para  $t = 1$  hasta  $T$  hacer
5      Para toda neurona  $C_i$  hacer
6          Calcular el grado de activación de  $A_i^{(t+1)}$ 
7      Fin.
8  Fin.
9  Para toda decisión  $d_k$  hacer
10      $\vartheta(k) \leftarrow A_i^{(T)}$ 
11  Fin.
```

---

### 2.5 Diseño e implementación de las FRCN sobre la plataforma Weka

La nueva propuesta ha sido especialmente diseñada para problemas de clasificación, por lo que se propone una implementación del modelo, incorporada a la plataforma Weka como un nuevo clasificador.

El proyecto Weka es un entorno de software para la experimentación y prueba de diferentes algoritmos de Aprendizaje Automatizado que se desarrolla en la Universidad de Waikato, Nueva Zelanda (Witten & Frank, 2005). Esta plataforma libre implementa, en el lenguaje de programación Java, una gran variedad de métodos estadísticos y de Inteligencia Artificial, así como técnicas para comparar sus resultados. Ello posibilita, que ante un problema dado que requiere aprendizaje automatizado, cualquier investigador pueda experimentar “en lote” con varios de estos métodos y determinar con cuáles se obtienen los mejores resultados.

Weka proporciona varios algoritmos de aprendizaje automatizado que pueden ser utilizados fácilmente; por ejemplo: se puede preprocesar el conjunto de datos, introducirlos en un esquema de aprendizaje y analizar el resultado de clasificación y su eficiencia. Además posee un sistema de interfaces gráficas de usuario que permiten la exploración de los datos (Explorer) y la experimentación (Experimenter) con los diversos algoritmos implementados.

### 2.5.1 Incorporación de un nuevo clasificador a la plataforma Weka

Por ser de código abierto, Weka es además extensible. El desarrollo de mejores extensiones a la plataforma Weka ha dejado de concentrarse en el lugar donde fue creado, ya que tiene una estructura de paquetes bien organizada y facilita a otros programadores la tarea de agregar modelos y algoritmos, o modificar los ya existentes de forma ordenada. Los paquetes principales son los siguientes (Guevara, 2007):

- ***associations***: contiene las clases que implementan los algoritmos de asociación.
- ***attributeSelection***: contiene las clases que implementan técnicas de selección de atributos.
- ***classifiers***: agrupa todas las clases que implementan algoritmos de clasificación y éstas a su vez se organizan en subpaquetes de acuerdo al tipo de clasificador.
- ***clusterers***: contiene las clases que implementan algoritmos de agrupamiento.
- ***core***: paquete central que contiene las clases controladoras del sistema. Es usado en la mayoría de las clases existentes. Las clases principales de este paquete son: ***Attribute***, ***Instance***, e ***Instances***.
- ***datagenerators***: paquete que contiene clases útiles en la generación de conjuntos de datos atendiendo al tipo de algoritmo que será usado.
- ***estimators***: contiene las clases que realizan estimaciones (generalmente probabilísticas) sobre los datos.
- ***experiment***: agrupa las clases controladoras que permiten la realización de experimentos con varias bases y diferentes algoritmos.

- **filters:** está constituido por las clases que implementan algoritmos de preprocesamiento.
- **gui:** contiene todas las clases que implementan la interfaz visual con el usuario.

Para añadir un nuevo clasificador en Weka se debe conocer los métodos imprescindibles que deben ser implementados, así como las clases y estructuras que Weka posee para el trabajo con los mismos. Además es necesario seguir cierta metodología descrita en (Bonet, 2008; Guevara, 2007) para la incorporación de un nuevo clasificador, con el objetivo de lograr una total compatibilidad con la herramienta y su óptimo funcionamiento.

Weka tiene implementada muchas facilidades de magnitudes de salidas y evaluación como la validación cruzada. A la hora de implementar un nuevo clasificador es importante tener en cuenta la clase abstracta *AbstractClassifier*, que es la que se utiliza para dar todas estas facilidades. Esta clase, que es la más importante en el paquete *classifiers* y constituye una superclase de todos los clasificadores existentes (Guevara, 2007), ya tiene definido los principales métodos que debe tener un clasificador, algunos de ellos deben ser redefinidos de acuerdo al objetivo que persiga el algoritmo de aprendizaje a implementar. De los métodos siguientes es estrictamente necesario redefinir el primero mientras basta definir al menos uno de los dos últimos:

- **buildClassifier():** se encarga de la construcción del modelo del clasificador tomando como parámetro las instancias de entrenamiento. Debe inicializar todas las variables que correspondan a las opciones específicas del esquema. Nunca debe modificar ningún valor de las instancias. Después de terminada la ejecución de este método el clasificador debe ser capaz de predecir la clase de cualquier instancia nueva.
- **classifyInstance():** permite clasificar una instancia concreta. Devuelve la clase en la que se ha clasificado o “desconocido” si no se consigue clasificar. Este algoritmo está diseñado de manera general también para regresión, de forma que en este caso devuelve un valor continuo.

- ***distributionForInstance()***: devuelve un vector con las probabilidades de pertenencia de la instancia a cada una de las clases.

Una vez implementados los métodos esenciales para el funcionamiento del clasificador, es necesaria la implementación de otros métodos auxiliares para facilitar el trabajo con la interfaz del usuario. En particular se deben modificar los métodos ***listOptions()***, ***setOptions()*** y ***getOptions()*** para que sea posible establecer los parámetros del entrenamiento por parte del usuario.

## **2.6 Conclusiones parciales**

La adecuada selección del umbral de similaridad resulta un aspecto sensible para el esquema de las RCN y su estimación requiere de la intervención de un experto del conocimiento o bien de un proceso computacional que puede llegar a ser costoso. En el capítulo se describió una nueva propuesta que incluye el enfoque difuso al esquema y elimina el umbral de similaridad. Además, se presentó la implementación del modelo propuesto para la plataforma Weka como un nuevo clasificador.

## **CAPÍTULO 3. EVALUACIÓN DEL MODELO**

En este capítulo se realizan múltiples experimentos y simulaciones con el objetivo de evaluar la capacidad predictiva y el rendimiento de la propuesta descrita en el capítulo anterior. Primero, se determinan los parámetros para los cuales el modelo en la predicción obtiene su mejor desempeño. Posteriormente, se compara la capacidad de predicción de las FRCN contra el modelo original, 18 clasificadores tradicionales y cinco clasificadores difusos<sup>1</sup> a través de 71 bases de conocimiento tradicionales para problemas de clasificación.

### **3.1 Selección de la relación de similaridad, t-norma e implicador difuso**

El objetivo principal de este epígrafe es el ajuste del modelo tal que alcance su mejor desempeño en la predicción. Para lograrlo, primero se realiza un análisis comparativo de las combinaciones de operadores posibles a utilizar en el cálculo de las aproximaciones y las reglas de activación. Luego, se prueban múltiples métricas para definir la relación de similaridad más apropiada.

#### **3.1.1 Selección de un implicador y dos t-normas difusas**

Como se ha mencionado, para determinar las aproximaciones y el grado de activación inicial en el modelo propuesto resulta necesario definir que implicador y t-normas utilizar. Para encontrar la combinación de operadores que mejor ajuste el modelo se realiza un análisis comparativo entre todas las combinaciones de operadores posibles a través de varios experimentos. En la Tabla 1 y la Tabla 2 se reportan las t-normas difusas (Wierman, 2010, p. 85) y los implicadores difusos (Wierman, 2010, p. 229) en los cuales se apoyan las comparaciones. Cada combinación de operadores comprende un implicador y dos t-normas difusas, necesarios para definir las aproximaciones inferiores y superior, y las reglas que determinar el grado de activación inicial de las neuronas del mapa.

---

<sup>1</sup> Algoritmos de aprendizaje que incluyen algún enfoque difuso en su modelo.

Tabla 1. T-normas comunes

Nombre	T-norma
Intersección estándar	$T(x, y) = \min\{x, y\}$
Producto algebraico	$T(x, y) = x * y$
Diferencia de límites (Lukasiewics)	$T(x, y) = \max(x + y - 1, 0)$
Producto drástico	$T(x, y) = \begin{cases} x, & y = 1 \\ y, & x = 1 \\ 0, & e. o. c \end{cases}$

Tabla 2. Implicadores difusos comunes

Implicador	Definición de $x \Rightarrow y$
Estándar	$I(x, y) = \begin{cases} 1, & x \leq y \\ 0, & x > y \end{cases}$
Kleene-Dienes	$I(x, y) = \max\{1 - x, y\}$
Lukasiewics	$I(x, y) = \min\{1 - x + y, 1\}$
Zadeh	$I(x, y) = \max\{1 - x, \min\{x, y\}\}$
Gödel	$I(x, y) = \begin{cases} 1, & x \leq y \\ y, & x > y \end{cases}$
Larsen	$I(x, y) = x * y$
Mamdani	$I(x, y) = \min\{x, y\}$
Reichenbach	$I(x, y) = 1 - x + x * y$
Yager	$I(x, y) = \begin{cases} 1, & x = y = 0 \\ y^x, & e. o. c \end{cases}$
Goguen	$I(x, y) = \begin{cases} 1, & x \leq y \\ \frac{y}{x}, & e. o. c \end{cases}$

La Figura 6 muestra el promedio de la precisión alcanzada por el modelo para cada una de las combinaciones. La Figura 6 (a) muestra los resultados de la experimentación utilizando la t-norma estándar en el cálculo de las reglas de activación, y las distintas combinaciones de

implicadores y t-normas en el cálculo de las aproximaciones inferior y superior. La Figura 6 (b), (c) y (d) muestra los mismos resultados, pero utilizando en el cálculo del grado de activación inicial de las neuronas la t-norma del producto algebraico, la t-norma de Lukasiewicz y la t-norma del producto drástico respectivamente.

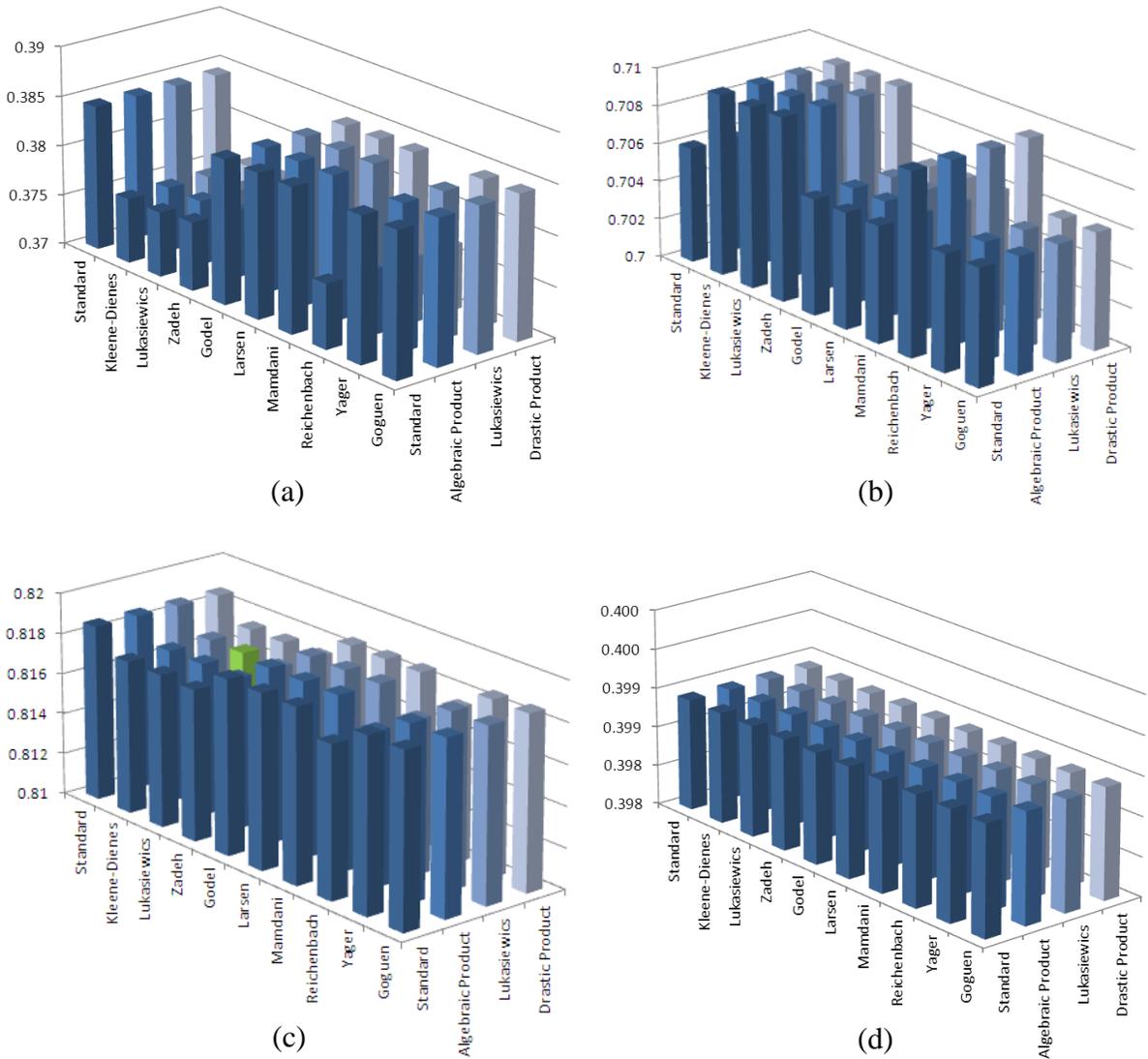


Figura 6. Precisión del modelo para cada una de las combinaciones

De los resultados obtenidos se puede concluir lo siguiente: la selección adecuada de la t-norma a utilizar en las reglas de activación es determinante en la precisión final del modelo; el modelo alcanza sus mejores resultados utilizando la t-norma de Lukasiewicz en el cálculo de los grados de excitación iniciales, y utilizando el implicador y la t-norma de Lukasiewicz para

determinar las aproximaciones inferior y superior. La mejor precisión alcanzada por el modelo fue de 0.8175.

### 3.1.2 Selección de la función de similitud

Un aspecto a ser cuidadosamente considerado cuando se diseña una relación de similitud es la adecuada selección de la función de similitud. En esta sección se realiza un análisis entre varias métricas, para determinar la función de similitud que mejor se adapte al modelo. Las funciones de similitud son construidas a partir de: Heterogeneous Manhattan-Overlap Metric (HMOM) (Silva & Batista, 2009, pp. 99–101) y Heterogeneous Euclidean-Overlap Metric (HEOM) (Wilson & Martinez, 1997).

La Figura 7 muestra el desempeño alcanzado por el modelo en la predicción para las funciones, basado en el promedio de instancias bien clasificadas. De estos resultados podemos concluir que el modelo posee un mejor desempeño utilizando la función HMOM (0.8175) en la construcción de la relación de similitud.

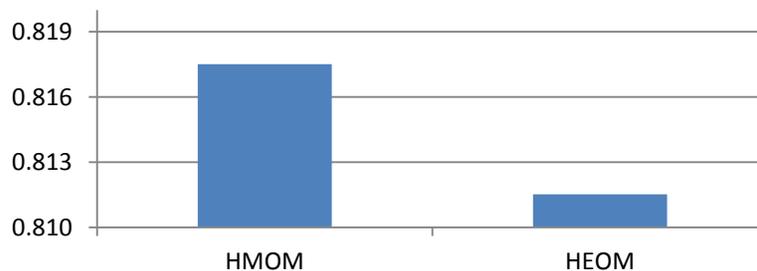


Figura 7. Precisión de las funciones de similitud

Con el objetivo de determinar si es estadísticamente significativa la superioridad de HMOM en comparación a HEOM se aplica una prueba de Wilcoxon (Wilcoxon, 1945). Esta prueba no paramétrica determina si dos muestras representan la misma población, en cuyo caso se acepta la hipótesis  $H_0$ . Esta prueba sugiere rechazar la hipótesis nula  $H_0$  ( $p\text{-value} = 0.438E - 3 < 0.05$ ) utilizando un intervalo de confianza del 95%, lo cual sugiere que existen diferencias significativas entre las funciones de similitud. Por lo que se puede concluir que la precisión alcanzada por la función de distancia HMOM es superior.

### 3.2 Análisis del desempeño del modelo

En esta sección se estudiará el desempeño de las FRCN. Con el objetivo de llevar a cabo un análisis empírico, la experimentación se apoya en 71 bases de casos clásicas tomadas de los repositorios de Keel y UCI Machine Learning. La Tabla 3 describe las propiedades (el número de instancias, características y clases de decisión) de las bases de casos usadas en la experimentación. Las bases de casos involucran diferentes características y permiten la evaluación de la capacidad de predicción de los modelos de aprendizaje automatizados. Las instancias con valores perdidos fueron eliminadas de las bases de casos, debido a que el modelo propuesto no soporta las bases de casos con esas características.

Los algoritmos de aprendizaje automatizado adoptados en la comparación incluye: Decision Table (DT) (Kohavi, 1995), Conjunctive Rule (CR) (King, 1966), Naive Bayes classifier (NB) (Luengo, García, & Herrera, 2009), Naive Bayes Updateable (NBU) (John & Langley, 1995), Support Vector Machines (SVM) (Keerthi, Shevade, Bhattacharyya, & Murthy, 2001), Multilayer Perceptron (MLP) (Hecht-Nielsen, 1989), Radial Basis Network (RBN) (Broomhead & Lowe, 1988), Simple Logistic (SL) (Sumner, Frank, & Hall, 2005), Decision Stump (DST) (Holte, 1993), Fast Decision Tree (REPT) (Su & Zhang, 2006), Decision Tree (J48) (Quinlan, 2014), Best-first Decision Tree (BFT) (Shi, 2007), Logistic Model Trees (LMT) (Landwehr, Hall, & Frank, 2005), Random Forests (RF) (Breiman, 2001), Random Tree (RT) (Aldous, 1991), Nearest-neighbor classifier (IB1) (Aha, Kibler, & Albert, 1991), K-nearest neighbors classifier (IBK) (Aha et al., 1991), K\* instance-based classifier (KSTAR) (Cleary, Trigg, & Others, 1995), Locally weighted learning (LWL) (Atkeson, Moore, & Schaal, 1997), Fuzzy K-Nearest Neighbours (FNN) (Aha et al., 1991), Fuzzy-Rough K-Nearest Neighbours (FRNN) (Jensen & Cornelis, 2008), Fuzzy Ownership K-Nearest Neighbours (FONN) (Aha et al., 1991), Vaguely-quantified K-Nearest Neighbours (VQNN) (Jensen & Cornelis, 2008) y Fuzzy Unordered Rule Induction (FURIA) (Huehn & Huellermeier, 2009). Nótese que se adoptan las variantes y los parámetros de configuración por defecto implementados en Weka para realizar los experimentos.

Tabla 3. Propiedades de las bases de casos adoptadas en la comparación

Base de casos	Instancias	Atributos	Clases	Base de casos	Instancias	Atributos	Clases	Base de casos	Instancias	Atributos	Clases	Base de casos	Instancias	Atributos	Clases
Appendicitis	106	8	2	ecoli-5an-nn	336	8	8	ionosphere	351	35	2	sonar	208	61	2
Australian	690	15	2	ecoli3	336	8	2	iris-10an-nn	150	5	3	spectfheart	267	45	2
balance-noise	625	5	3	german	1000	21	2	iris-20an-nn	150	5	3	tae	151	6	3
balance-scale	625	5	3	glass-0-1-2-3_vs_4-5-6	214	10	2	iris-5an-nn	150	5	3	vehicle	846	19	4
Banana	5300	3	2	glass-10an-nn	214	10	6	iris	150	5	3	vehicle0	846	19	2
Breast	277	10	2	glass-20an-nn	214	10	6	iris0	150	5	2	vehicle1	846	19	2
Bupa	345	7	2	glass-5an-nn	214	10	6	lymph	148	19	4	vehicle2	846	19	2
Cleveland	297	14	5	glass	214	10	6	mammographic	830	6	2	vehicle3	846	19	2
contraceptive-noise	1473	10	3	glass0	214	10	2	movement_libras	360	91	15	vowel	990	14	11
credit-a	690	16	2	glass1	214	10	2	mushroom	5644	23	2	wine-5an-nn	178	14	3
credit-g	1000	21	2	glass2	214	10	2	new-thyroid	215	6	2	wine	178	14	3
Crx	653	16	2	glass3	214	10	2	pima-10an-nn	768	9	2	winequality-red	1599	12	6
Dermatology	358	35	6	glass6	214	10	2	pima-20an-nn	768	9	2	wisconsin	683	10	2
Ecoli	336	8	8	haberman	306	4	2	pima-5an-nn	768	9	2	yeast	1484	9	10
ecoli0	220	8	2	hayes-roth	160	5	3	pima-imbalanced	768	9	2	yeast1	1484	9	2
ecoli1	336	8	2	heart-5an-nn	270	14	2	pima	768	9	2	yeast3	1484	9	2
ecoli2	336	8	2	heart-statlog	270	14	2	saheart	462	10	2				
ecoli-0_vs_1	220	8	2	heart	270	14	2	segment	2310	20	7				

La Figura 8 muestra el desempeño alcanzado por los clasificadores en la predicción, basados en el promedio de instancias bien clasificadas en las bases de casos seleccionadas. Los resultados de las RCN respecto al mejor umbral de similaridad para cada una de las bases. De estos resultados podemos concluir que el clasificador LMT posee el mejor desempeño (0.8268), seguido por RF (0.8261), FRCN (0.8175), MLP (0.8165), VQNN (0.8154) y RCN (0.8148). Los peores resultados fueron obtenidos por DST (0.6813), CR (0.6836), FNN (0.7117) y LWL (0.7415).

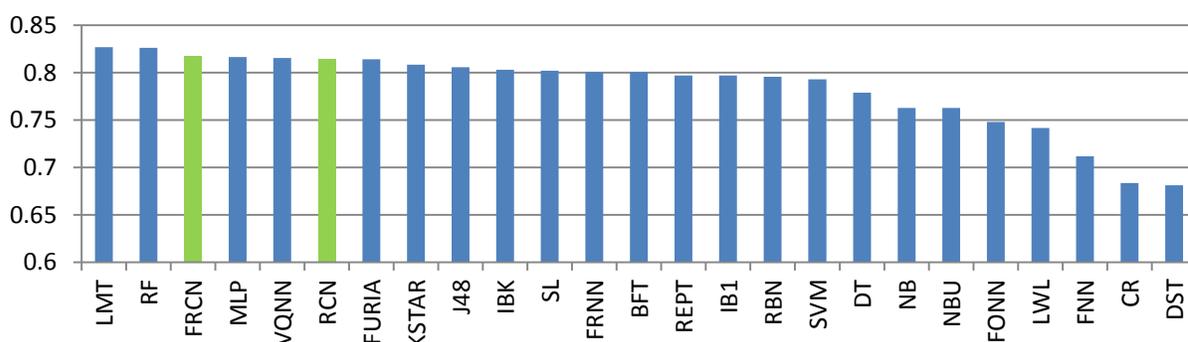


Figura 8. Precisión alcanzada por los clasificadores.

La Figura 9 se muestra el promedio de la precisión alcanzada por las RCN utilizando diferentes umbrales de similaridad entre 0.6 y 0.9 (el umbral similaridad predefinido empleado por las RCN es 0.9). En estos resultados se observa un incremento de la precisión a medida que crece el umbral. El umbral para el cual se obtiene un mejor desempeño es  $\xi = 0.99$  (0.8066), el cual no supera que la predicción alcanzada por las FRCN.

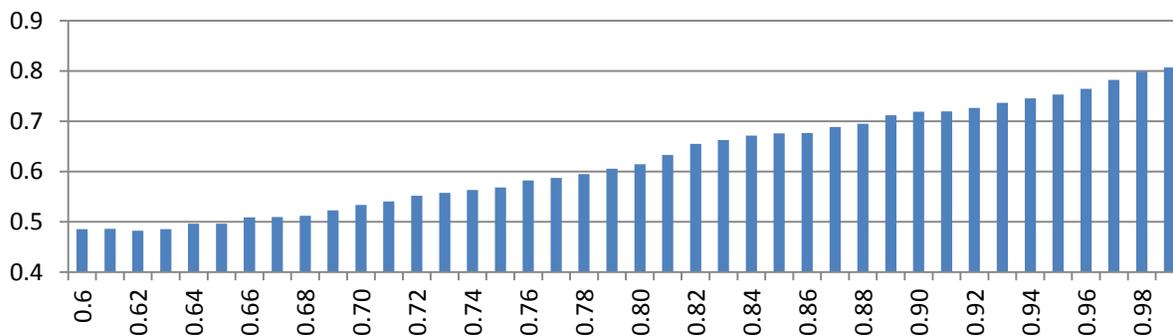


Figura 9. Precisión de las RCN para diferentes umbrales de similaridad

Con el objetivo de descubrir la existencia de diferencias significativas, se aplica una prueba Friedman (Friedman, 1937) para el análisis de la varianza. Friedman es un método estadístico no paramétrico para comparaciones de múltiples muestras relacionadas. Esta prueba sugiere rechazar la hipótesis nula  $H_0$  ( $p\text{-value} = 6.9398E - 53 < 0.05$ ) utilizando un intervalo de confianza del 95%, lo cual sugiere que existen diferencias significativas entre los algoritmos involucrados en el estudio. Por tal motivo, es necesario usar pruebas no paramétricas más específicas, que permitan comparar los algoritmos por pares. La Figura 10 muestra los niveles alcanzados por los clasificadores en la prueba de Friedman, donde las FRCN son el séptimo mejor modelo del ranking.

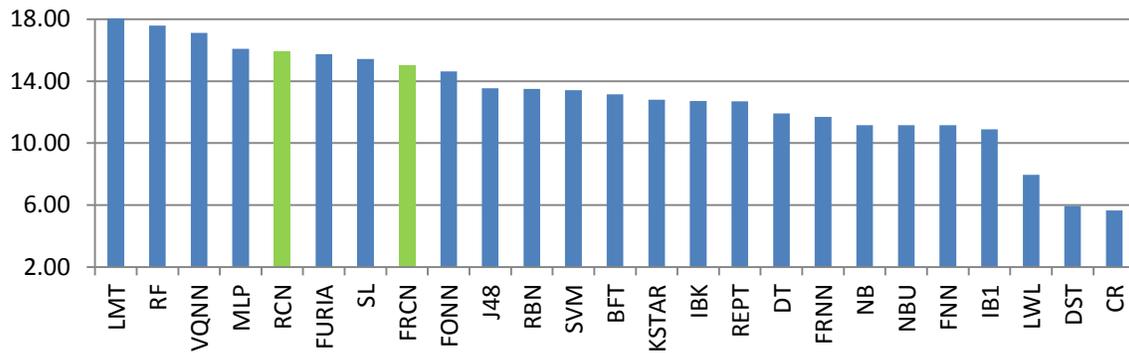


Figura 10. Ranking de clasificadores en la prueba de Friedman

El siguiente paso se centra en determinar si es estadísticamente significativa la superioridad de los clasificadores en comparación al modelo propuesto. Para el análisis del comportamiento de dos algoritmos concretos se recomienda utilizar la prueba de Wilcoxon (Wilcoxon, 1945). Esta prueba no paramétrica determina si dos muestras representan la misma población, en cuyo caso se acepta la hipótesis  $H_0$ . La Tabla 4 muestra los  $p$ -valores asociados, usando un nivel de significación de 0.05 referente al 95% del intervalo de confianza. Este experimento solo considera los pares de algoritmos que implican a las FRCN y las RCN, el resto de las combinaciones no aportan información relevante al estudio.

Tabla 4. Resultados de las pruebas de Wilcoxon

	$p$ -valores	Hipótesis $H_0$		$p$ -valores	Hipótesis $H_0$
FRCN vs RF	0.0030	Rechaza	RCN vs RF	0.0133	Rechaza
FRCN vs LMT	0.0562	Aceptada	RCN vs LMT	0.0617	Aceptada
FRCN vs MLP	0.6501	Aceptada	RCN vs MLP	0.9364	Aceptada
FRCN vs VQNN	0.1131	Aceptada	RCN vs VQNN	0.0881	Aceptada
FRCN vs FURIA	0.6402	Aceptada	RCN vs FURIA	0.9318	Aceptada
FRCN vs RCN	0.7418	Aceptada	RCN vs SL	0.4841	Aceptada
FRCN vs SL	0.5592	Aceptada			

Las pruebas de Wilcoxon solo identifican diferencias significativas a un nivel del 95% entre RF y el modelo propuesto y el original. Esto sugiere que la capacidad de predicción de las FRCN es comparable al modelo original y competitiva a los modelos del estado del arte actuales.

### 3.3 Conclusiones parciales

Durante el capítulo los experimentos fueron dirigidos en dos direcciones. Primero, se estudió el desempeño de las FRCN para diferentes combinaciones de operadores y relaciones de similaridad con el objetivo de definir los parámetros que mejor ajusten el modelo. Los resultados numéricos mostraron que los operadores de Lukasiewicz son los más adecuados para definir las aproximaciones y las reglas de activación en la nueva propuesta; y la relación de similaridad que mejor precisión alcanza es construida a partir de la función de distancia HMOM. Finalmente, los experimentos comparativos realizados para evaluar la capacidad predictiva y el rendimiento de la propuesta descrita sugieren que la capacidad de predicción de las FRCN es comparable al modelo original y competitiva a los modelos del estado del arte actuales.

## CONCLUSIONES

Las RCN son un modelo granular diseñado especialmente para problemas de clasificación donde las bases de conocimientos manifiesten inconsistencias. Este modelo emplea una relación de inseparabilidad “débil” para lidiar con la similaridad entre objetos. La selección adecuada del umbral de similaridad es un aspecto sensible para el algoritmo y su estimación requiere de la intervención de un experto del conocimiento o bien de un proceso computacional que puede llegar a ser costoso, siendo este el problema de investigación abordado en este estudio. Para lidiar con este problema se propuso una nueva alternativa que incluye el enfoque difuso al esquema y elimina el umbral de similaridad. Además, se presentó la implementación del modelo propuesto para la plataforma Weka como un nuevo clasificador.

Para definir los parámetros que mejor ajustan el nuevo modelo se realizaron múltiples experimentos, los cuales mostraron que los operadores de Lukasiewicz son los más adecuados para definir las aproximaciones y las reglas de activación en la propuesta.

Finalmente, los experimentos comparativos realizados para evaluar la capacidad predictiva y el rendimiento de la propuesta descrita sugieren que la capacidad de predicción de las FRCN se comparable al modelo original y competitiva a los modelos del estado del arte actuales.

## **RECOMENDACIONES**

1. Extender la topología de las FRCN para que no pierdan la semántica de la decisión de abstención del modelo de decisión de tres alternativas.
2. Evaluar la precisión del modelo utilizando otras funciones de similaridad en la construcción de la relación de similaridad.
3. Extender la experimentación del algoritmo de aprendizaje discutido para otras bases de conocimientos.

## BIBLIOGRAFÍA

- Abraham, A., Falcon, R., & Bello, R. (2009). *Rough Set Theory: a True Landmark in Data Analysis*. Berlin-Heidelberg, Germany: Springer Verlag.
- Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37–66.
- Aldous, D. (1991). The continuum random tree. I. *The Annals of Probability*, 1–28.
- Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). Locally weighted learning for control. In *Lazy learning* (pp. 75–113). Springer.
- Bello, R., Falcon, R., Pedrycz, W., & Kacprzyk, J. (2008). *Granular Computing: at the Junction of Rough Sets and Fuzzy Sets*. Berlin-Heidelberg, Germany: Springer Verlag.
- Bello, R., & Garcia, M. M. (2014). Probabilistic Approaches to the Rough Set Theory and Their Applications in Decision-Making. In R. Espin, R. Bello, A. Cobo, J. Marx, & A. Racet (Eds.), *Soft Computing for Business Intelligence* (pp. 67–80). Springer Berlin Heidelberg.
- Bello, R., & Verdegay, J. L. (2012). Rough sets in the Soft Computing environment. *Information Sciences*, 212, 1–14. <http://doi.org/10.1016/j.ins.2012.04.041>
- Bonet, I. (2008). *Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial*. Universidad Central “Marta Abreu” de Las Villas.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Broomhead, D. S., & Lowe, D. (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks*.
- Bueno, S., & Salmeron, J. L. (2009). Benchmarking main activation functions in fuzzy cognitive maps. *Expert Systems with Applications*, 36(3), 5221–5229. <http://doi.org/10.1016/j.eswa.2008.06.072>
- Cleary, J. G., Trigg, L. E., & Others. (1995). K\*: An instance-based learner using an entropic distance measure. In *Proceedings of the 12th International Conference on Machine learning* (Vol. 5, pp. 108–114).
- Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200), 675–701.
- Guevara, L. (2007). *Extensión del sistema Weka con la incorporación de Redes Neuronales Recurrentes*. Universidad Central de Las Villas.
- Hecht-Nielsen, R. (1989). Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on* (pp. 593–605).

- Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1), 63–90.
- Huehn, J. C., & Huellermeier, E. (2009). FURIA: An Algorithm for Unordered Fuzzy Rule Induction. *Data Mining and Knowledge Discovery*.
- Inuiguchi, M., Wu, W.-Z., Cornelis, C., & Verbiest, N. (2013). Fuzzy-rough Hybridization.
- Jensen, R., & Cornelis, C. (2008). A New Approach to Fuzzy-Rough Nearest Neighbour Classification. In *6th International Conference on Rough Sets and Current Trends in Computing* (pp. 310–319).
- Jensen, R., & Shen, Q. (2009). New approaches to fuzzy-rough feature selection. *IEEE Transactions on Fuzzy Systems*, 824–838.
- John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (pp. 338–345).
- Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13(3), 637–649.
- King, W. L. (1966). Learning and utilization of conjunctive and disjunctive classification rules: A developmental study. *Journal of Experimental Child Psychology*, 4(3), 217–231. [http://doi.org/http://dx.doi.org/10.1016/0022-0965\(66\)90022-1](http://doi.org/http://dx.doi.org/10.1016/0022-0965(66)90022-1)
- Kohavi, R. (1995). The power of decision tables. In *Machine Learning: ECML-95* (pp. 174–189). Springer.
- Kosko, B. (1986). Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24(1), 65–75. [http://doi.org/10.1016/S0020-7373\(86\)80040-2](http://doi.org/10.1016/S0020-7373(86)80040-2)
- Kosko, B. (1988). Hidden patterns in combined and adaptive knowledge networks. *International Journal of Approximate Reasoning*, 2(4), 377–393. [http://doi.org/10.1016/0888-613X\(88\)90111-9](http://doi.org/10.1016/0888-613X(88)90111-9)
- Kosko, B. (1992). *Neural Networks and Fuzzy systems, a dynamic system approach to machine intelligence*. Prentice-Hall, Englewood Cliffs.
- Landwehr, N., Hall, M., & Frank, E. (2005). Logistic model trees. *Machine Learning*, 59(1-2), 161–205.
- Luengo, J., García, S., & Herrera, F. (2009). A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests. *Expert Systems with Applications*, 36(4), 7798–7808.
- Miao, Y., & Liu, Z.-Q. (2000). On causal inference in fuzzy cognitive maps. *IEEE Transactions on Fuzzy Systems*, 8(1), 107–119.
- Nápoles, G., Grau, I., Falcon, R., Bello, R., & Vanhoof, K. (2015). A Granular Intrusion Detection System using Rough Cognitive Networks. In R. Abielmona, R. Falcon, N. Zincir-Heywood, & H. Abbass (Eds.), *Recent Advances in Computational Intelligence in Defense and Security*. Springer Verlag.

- Nápoles, G., Grau, I., Papageorgiou, E., Bello, R., & Vanhoof, K. (2016). Rough Cognitive Networks. *Knowledge-Based Systems*, *91*, 46–61.
- Nápoles, G., Grau, I., Vanhoof, K., & Bello, R. (2014). Hybrid Model Based on Rough Sets Theory and Fuzzy Cognitive Maps for Decision-Making. In M. Kryszkiewicz, C. Cornelis, D. Ciucci, J. Medina-Moreno, H. Motoda, & Z. Ras (Eds.), *Rough Sets and Intelligent Systems Paradigms* (pp. 169–178). Springer Berlin Heidelberg.
- Pawlak, Z. (1982). Rough sets. *International Journal of Computer {&} Information Sciences*, *11*(5), 341–356. <http://doi.org/10.1007/BF01001956>
- Quinlan, J. R. (2014). *C4.5: Programs for Machine Learning*. Retrieved from <http://books.google.com/books?hl=es{&}lr={&}id=b3ujBQAAQBAJ{&}pgis=1>
- Shi, H. (2007). *Best-first decision tree learning*. Citeseer.
- Silva, D. F., & Batista, G. E. A. P. A. (2009). How k-Nearest Neighbor Parameters Affect its Performance, (Asai), 95–106.
- Slowinski, R., & Vanderpooten, D. (2000). A generalized definition of rough approximations based on similarity. *IEEE Transactions on Knowledge {&} Data Engineering*, *12*, 331–336.
- Su, J., & Zhang, H. (2006). A Fast Decision Tree Learning Algorithm. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1* (pp. 500–505). AAAI Press. Retrieved from <http://dl.acm.org/citation.cfm?id=1597538.1597619>
- Sumner, M., Frank, E., & Hall, M. (2005). Speeding up logistic model tree induction. In *Knowledge Discovery in Databases: PKDD 2005* (pp. 675–683). Springer.
- Wierman, M. J. (2010). *An Introduction to the Mathematics of Uncertainty*.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, *1*, 80–83.
- Wilson, D. R., & Martinez, T. R. (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, *6*, 1–34.
- Witten, I. H., & Frank, E. (2005). *Data Mining Practical Machine Learning Tools and Techniques*. (J. Gray, Ed.) (Second Edi). Morgan Kaufmann Publishers.
- Yao, Y. (2010). Three-way decisions with probabilistic rough sets. *Information Sciences*, *180*(3), 341–353. <http://doi.org/10.1016/j.ins.2009.09.021>
- Yao, Y. (2011). The superiority of three-way decisions in probabilistic rough set models. *Information Sciences*, *181*(6), 1080–1096.