Ministerio de Educación Superior
Universidad Central "Marta Abreu" de Las Villas
Facultad de Matemática, Física y Computación

TRABAJO DE DIPLOMA

Implementación y Validación del Módulo Basado en Casos de un Modelo Híbrido

AUTOR

Yasel Couce Sardiñas

TUTORES

MSc. Yanet Rodríguez Sarabia
Dra. María Matilde García Lorenzo
Lic. Rafael Jesús Falcón Martínez

2005
Santa Clara

Hago constar que el presente trabajo fue realizado en la Universidad Central "Marta Abreu" de las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a 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 publicados sin 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 el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____
Firma del Tutor

_____
Firma del Jefe de Seminario
de Inteligencia Artificial

_____
Firma del Responsable de
Información Científico- Técnica

"La alegría está en la lucha, en el esfuerzo, en el sufrimiento que supone la lucha y no en la victoria."

Mahatma Gandhi

A mis padres porque son todo lo que yo quisiera llegar a ser algún día.

A mis hermanos y a la Churri, porque son la fuente de inspiración de mi vida.

A Osmedy González Ríos, el mejor de los estudiantes.

Agradezco a mi familia por todo su apoyo incondicional, a mis amigos por estar presente cuando más los necesité, a Oria por cuidarme y quererme tanto, a todo el grupo de IA y en especial a Yanet y a su familia por apoyarme hasta el final.

Resumen

En el grupo de Inteligencia Artificial se diseñó un modelo híbrido que combina Redes Neuronales Artificiales (RNA) y Razonamiento Basado en Casos (RBC), donde la RNA propone una solución a un problema y el RBC la justifica en el contexto de los casos más semejantes a este. El modelo de RNA implementado es asociativo, y por tanto en su topología se asocia cada valor presente para un rasgo en la base de casos a una neurona. En este sentido se extiende este modelo utilizando conjuntos borrosos, y así considerar los términos lingüísticos como los valores representativos en este tipo de rasgo.

Por otro lado, una muestra de aprendizaje para la RNA puede que no sea representativa del dominio del problema, y por tanto la RNA no lograría generalizaciones correctas. Otra situación pudiera ser que se presenten problemas, los cuales por sus características se considerarían excepciones del dominio y cuya solución por la RNA puede no ser correcta. En estos casos el enfoque híbrido del modelo original se pudiera extender, utilizando estos pesos de la RNA en una función de semejanza similar a la anterior, donde el RBC resuelva el problema.

En el presente trabajo se implementa una herramienta computacional para la componente basada en casos del nuevo modelo híbrido, en la cual el RBC desempeña los roles de justificador y resolvedor. La validación utilizando archivos de datos internacionales con la herramienta desarrollada, permite concluir la factibilidad de estas extensiones al modelo original.

Abstract

A hybrid model that combines Artificial Neural Networks (ANN) and Case-Based Reasoning (CBR) was designed at the Artificial Intelligence research group. In it, the ANN proposes a solution to a problem and the CBR justifies it in the context of the most similar cases to that one. The ANN implemented model is an associative one, and therefore in its topology each feature's value that appears in the cases base is associated with a neuron. In this sense the hybrid model is extended using fuzzy sets, considering linguistic terms as representative values for this type of features.

On the other hand, a learning sample for the ANN might not be representative of the problem's domain, and therefore the ANN would not obtain correct generalizations. Another situation could be that certain specific problems may appear -which could be considered by its characteristics as exceptions of the domain- and whose solution by the ANN could not be right. In these cases, the hybrid approach of the original model could be extended using these weights from ANN in a similarity function where the CBR solves the problem.

In this present work, a computational tool for the cases-based component of the new hybrid model is implemented where the CBR carries out both the rolls of justifier and solver. The validation using well-known international data files with the developed tool allows concluding the feasibility of these extensions to the original model.

Índice

Introducción

En el grupo de Inteligencia Artificial de la facultad de MFC se desarrolló un modelo híbrido [GAR96] que combina las Redes Neuronales Artificiales (RNA) y el Razonamiento Basado en Casos (RBC). La RNA resuelve el problema, es decir, ante la descripción de un problema valoriza los rasgos definidos como objetivos; y la componente basada en casos justifica la solución en el contexto de los *k* casos más similares a este, utilizando una función de similitud o semejanza (FS) que toma la información requerida de los pesos de la RNA. De esta forma se aprovechan las ventajas que las RNA tienen en cuanto a la capacidad de aprendizaje y se elimina su limitación de no permitir explicar la solución encontrada. Este modelo fue implementado en la herramienta *SISI*.

El modelo de RNA que implementa es del tipo asociativo, lo cual posibilita resolver problemas variando la definición de los rasgos como predictores u objetivos, sin necesidad de calcular nuevamente los pesos. En la topología de estos modelos se asocia una neurona a cada valor posible del rasgo. Sin embargo, en la mayoría de las ocasiones no es necesario considerar todos los valores que aparecen en la BC, y basta con tomar valores que puedan ser representativos de todo un grupo de valores próximos a él.

Ideas preliminares de la extensión de este modelo para representar los rasgos lineales (continuos o discretos) utilizando conjuntos borrosos, se presenta en [GAR00]. En este nuevo modelo los rasgos lineales se representan por los términos lingüísticos definidos. Esta extensión del modelo original requiere ser validada previo a su extensiva aplicación a problemas reales. Para esto se requiere de una herramienta computacional que implemente el nuevo modelo con facilidades a tales efectos.

Además, aunque a capacidad de una RNA para generalizar puede producir mejores resultados que el CBR, esto requiere que sea entrenada con una significativa cantidad de datos representativos de todas las situaciones que se puedan presentar en el dominio de aplicación. En la aplicación a un problema real donde inicialmente no se tengan suficientes ejemplos el RBC pudiera ser más efectivo que una RNA. Luego, como resultado de ir incorporando problemas resueltos y apropiados a la BC, la RNA pudiera ser reentrenada como se plantea en [LEE02]. También se pueden presentar problemas que por sus características, aunque la RNA haya logrado hacer generalizaciones, son excepciones del dominio de aplicación cuya solución no se ajusta a la generalidad de los casos. En tales situaciones el RBC podría lograr mejor desempeño que una

RNA. Teniendo en cuenta estas reflexiones surge la idea de extender el modelo original añadiendo una nueva funcionalidad, la de utilizar el RBC para resolver un problema.

Es por ello que el presente trabajo tiene el objetivo general siguiente:

Implementar y validar el módulo basado en casos de un modelo híbrido

Y los objetivos específicos que a continuación se relacionan:

1. Implementar una herramienta para el módulo basado en casos, a partir de la extensión del modelo del *SISI* utilizando conjuntos borrosos.
2. Validar la extensión de la función de semejanza implementada en *SISI*, al tratamiento de rasgos continuos utilizando conjuntos borrosos.
3. Modificar el módulo basado en casos implementado en *SISI*, para proponer una solución con el RBC.

Capítulo 1: Los sistemas híbridos basados en el conocimiento

Los Sistemas Basados en el Conocimiento (SBC) se conforman a partir de tres componentes básicas: la base de conocimiento, la máquina de inferencia y la interfaz. Adicionalmente se pueden añadir otros módulos que facilitan el autoaprendizaje de la base de conocimiento, el poder dar explicaciones a las soluciones encontradas, así como manipular la incertidumbre.

Un sistema de razonamiento tiene varias componentes básicas, entre ellas el esquema o forma de representación del conocimiento (lógica, *frames*, redes semánticas, etc.), un conjunto de reglas de inferencia (resolución, instanciación de los *frames*, etc.), y algunos medios para controlar la forma en que el sistema aplica las reglas de inferencia en buscar una solución (razonamiento hacia atrás, razonamiento hacia adelante, etc.).

Una vez seleccionada la clase de razonamiento (hacia atrás, hacia adelante, etc.) durante el proceso de búsqueda es necesario definir la estrategia con la que se explora el espacio, la cual puede ser una búsqueda a ciegas (no se usan las propiedades internas de los subproblemas y los subobjetivos dentro de estos subproblemas) en la que se usan métodos como la búsqueda primero a lo ancho o primero en profundidad, o una búsqueda heurística.

Existen varios tipos de SBC, dependiendo de la forma de representar el conocimiento y del razonamiento que se emplee. A continuación nos detendremos en algunas de las tecnologías que se emplean en Inteligencia Artificial para desarrollar SBC.

1.1 Tendencias actuales en el desarrollo de SBC

El concepto de *Soft Computing* (SC) comenzó a cristalizarse durante los últimos años. A partir de las dificultades mostradas por la Inteligencia Artificial, basada principalmente en lógica de predicados y técnicas de manipulación de símbolos, para construir máquinas las cuales pudieran ser llamadas inteligentes en el sentido de tener éxito en aplicaciones reales se han desarrollado una colección de metodologías con ese propósito y que hoy se agrupan en el concepto de SC. La esencia de la SC es que ella se acomoda a la imprecisión del mundo real. La guía principal de la SC es explotar la tolerancia a la imprecisión, la incertidumbre y la verdad parcial para alcanzar tratabilidad, robustez y soluciones de bajo costo. Las RNA, la teoría de los conjuntos borrosos, y

los Algoritmos Genéticos (AG) se consideran las raíces de la SC o inteligencia computacional como también fue denominada a inicio de la década del 90 [SAN01].

Estas tres áreas se complementan entre sí, y tienen en común requerimientos de equipamiento computacional moderno y que comparten el mismo cuello de botella que es la adquisición del conocimiento necesaria para su funcionamiento. Por ello recientemente a este grupo se ha añadido el aprendizaje automatizado, y se espera que poco a poco se vayan incorporando otras áreas de la Inteligencia Artificial. En este contexto la más reciente inclusión ha sido el RBC.

1.1.1 La Redes Neuronales Artificiales

Las RNA no son más que un modelo artificial y simplificado del cerebro humano, que es el ejemplo más perfecto del que disponemos para un sistema que es capaz de adquirir conocimiento a través de la experiencia. Una RNA es "un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona".

Debido a su constitución y a sus fundamentos, las RNA presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas.

Los conceptos básicos de los que consta una RNA, expuestos en [MAT01], son:

- Neurona: Cualquier modelo de RNA consta de estos dispositivos elementales de proceso. Se pueden encontrar, generalmente, tres tipos: aquellas que reciben estímulos externos que tomarán la información de entrada, y se determinan neuronas sensoras; las que reciben esta información de entrada y generan cualquier tipo de representación interna de la información, que se les conoce como neuronas de preprocesamiento; y las unidades de salida, cuya misión es dar la respuesta del sistema que también son llamadas unidades externas.
- Peso: Las conexiones entre las neuronas son las que determinan el estado de activación de las mismas. Para medir la fortaleza de una conexión o enlace entre dos neuronas i y j se utiliza una medida numérica, que es lo que conocemos por peso (w_{ij}). Si el valor de este

peso es positivo, se dice que la conexión es excitadora; si es negativo, estamos en presencia de una conexión inhibidora; y la ausencia de enlace se denota haciendo $w_{ij} = 0$.

- Función de entrada: Es la forma que tiene la neurona de combinar todos los valores de entrada que le llegan –procedentes del exterior o de otras neuronas- en uno solo, cuantificado por los pesos de dichos enlaces. También se le conoce como entrada neta, y existen múltiples formas de cálculo. La idea siempre es combinar la entrada proveniente de cada neurona con el peso del enlace con ella, que se denomina entrada neta.
- Función de activación: Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un "estado de activación". Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado. La función de activación calcula el estado de actividad de una neurona; transformando la entrada global (menos el umbral, θ_i) en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (–1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 o –1) o activa (1).
- Función de salida: El último componente que una neurona necesita es la función de salida. El valor resultante de esta función es la salida de la neurona i (out_i); por ende, la función de salida determina qué valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona subsiguiente. Normalmente, no cualquier valor es permitido como una entrada para una neurona, por lo tanto, los valores de salida están comprendidos en el rango [0, 1] o [-1, 1]. También pueden ser binarios {0, 1} o {-1, 1}.

Desde el surgimiento de las RNA se han desarrollado diversos modelos orientados a resolver problemas de agrupamiento, clasificación y búsqueda asociativa de información [HIL95] [ZUR92]. En [GAR03] se muestra cómo utilizar las RNA en la búsqueda de información, en la cual esta se puede utilizar para almacenar la información y luego realizar búsquedas en ella, o como una fuente de información auxiliar para hacer recuperaciones a partir de información parcial (patrón de búsqueda) en una base de datos. A continuación describiremos brevemente el modelo Activación Interactiva y Competencia (IAC), que es útil en tales aplicaciones.

En la topología de este modelo las neuronas se distribuyen en grupos y se establecen enlaces entre las neuronas de grupos diferentes y del mismo grupo. Las conexiones entre las neuronas de

un mismo grupo son excitadoras, mientras que las conexiones dentro de los grupos son inhibidoras. La esencia del modelo es que las neuronas de grupos diferentes tratan de excitarse mutuamente de modo que cada unidad trata de incrementar el nivel de activación de sus unidades adyacentes en otros grupos, y dentro de cada grupo se establece una competencia en la cual cada neurona trata de disminuir el nivel de activación de sus compañeras de grupo.

En la etapa de explotación de esta RNA las unidades de procesamiento cambian su activación considerando su activación actual, la entrada desde otras neuronas (del mismo grupo y desde otros grupos), y la entrada exterior a la red. En los cálculos que se realizan durante este proceso intervienen varios parámetros, que casi siempre son iguales para todas las neuronas de la red, y que se mencionan continuación:

- Max: valor máximo de la activación.
- Min: valor mínimo de la activación.
- Estr: influencia de las entradas externas a la red.
- α: escala de la fortaleza de las entradas excitadoras a las neuronas desde otras neuronas.
- γ: escala de la fortaleza de las entradas inhibidoras a las neuronas desde otras neuronas.
- Rest: nivel de activación de reposo al cual tienden las activaciones en ausencia de entradas internas.
- Decay: fortaleza de la tendencia al nivel de reposo.

Estos parámetros también deben ser estimados durante el entrenamiento de la red para cada problema, y de ello depende su buen funcionamiento. Ideas de cómo influye la variación de estos parámetros en la respuesta de este modelo de red se presenta en [ROD02].

Las RNA son convenientes para analizar grandes cantidades de datos y establecer patrones y características en situaciones donde las reglas no son conocidas. Además su desempeño, con respecto a otras tecnologías, no se afecta tanto al trabajar en dominios donde haya información incompleta o ruidosa.

1.1.2 Los Sistemas Basados en casos

El Razonamiento Basado en Casos es un enfoque para resolver problemas basado en la recuperación de casos semejantes a la descripción del problema a resolver, y la adaptación de las soluciones. La tecnología de los Sistemas Basados en Casos (SBCA) tiene mucho en común con

la solución de problemas por analogía, la búsqueda asociativa de información y los métodos de aprendizaje automatizado.

Este término razonamiento se refiere en general a diferentes clases de actividad, entre ellas: extraer conclusiones desde un conjunto de hechos, diagnosticar posibles causas para alguna situación, resolver un problema, entre otras. Razonar incluye inferencias pequeñas organizadas en función de un objetivo o problema principal, aunque ambos términos se emplean frecuentemente indistintamente. Los esquemas de razonamiento se pueden clasificar atendiendo a los criterios siguientes:

a) Precisión.

En un extremo de esta clasificación están los procesos de razonamiento que son lógicamente válidos, es decir, aquellos en los que el resultado del razonamiento es siempre verdadero en cualquier interpretación en la cual las premisas sean verdaderas. En el otro extremo están los sistemas de razonamiento borrosos.

b) Nivel.

Una segunda clasificación se puede realizar sobre la base del nivel al cual un programa razona: razonar dentro del dominio del problema en sí o razonar sobre el problema (meta razonamiento). El primer caso se refiere al empleo del conocimiento y los operadores de búsqueda para resolver un problema y en el otro el razonamiento se realiza para dirigir la exploración en el espacio de búsqueda.

c) Generalidad.

En un extremo de esta clasificación están los demostradores de teoremas de propósito general y en el otro los sistemas cuyos esquemas de representación, reglas de inferencia y estrategias de búsqueda, han sido optimizadas para un dominio particular.

El RBC representa un nuevo paradigma de búsqueda pues el formalismo que se utiliza sirve para representar no conocimiento explícito, sino ejemplos de problemas resueltos del dominio de aplicación. Estos sistemas son particularmente apropiados en dominios donde el proceso de adquisición del conocimiento mediante reglas se dificulta. Ellos poseen características presentes en los humanos al resolver un problema, y que son difíciles de simular usando la lógica, las técnicas analíticas de SBC y las tecnologías del *software* estándar.

Los casos, según [GUA94] representan situaciones experimentadas previamente. Generalmente un caso contiene la descripción del problema o situación; la solución dada al problema; si existe realimentación en la ejecución: el resultado de la ejecución, si ocurrió algún error en la

predicción: explicación de las anomalías, estrategia de reparación y referencia al próximo resultado. Otra alternativa es presentada por Jurísica en [JUR93] y es que el caso contenga toda la secuencia del proceso de solución al problema, representando los episodios ocurridos en este proceso. Un caso es su contenido y el contexto en el cual es aplicable. Se considera que un caso puede representar una situación completa (plan, diseño); la cual puede ser recuperada, adaptada y reusada posteriormente, eliminando la necesidad de descomponer y recomponer.

Considerando su forma, un caso puede ser un simple concepto o un conjunto conectado de subcasos; este puede ser una instancia específica o una generalización. Se han propuesto muchas y diferentes representaciones para un caso entre las que se encuentran: representar un caso como una simple lista de rasgos booleanos; como unidades de información de complejidad variable pero bien delimitada, mediante conjuntos de nodos de una red que se pueden compartir por varios casos; como subpartes de una gran jerarquía, entre otras representaciones.

De forma común a todas estas representaciones está el problema de cómo organizar los casos en la base, organización que dependerá de la representación dada a los casos y que se encuentra estrechamente ligada al proceso de recuperación de casos semejantes, y que puede incidir en la eficiencia del sistema.

1.1.3 Los Sistemas de Inferencia Borrosos

La idea de los conjuntos borrosos [ZAD76] viene de la siguiente observación: las clases de objetos en la vida diaria no tienen límites bien definidos. La fuente de imprecisión es la ausencia de criterios definidos rigurosamente de membresía a clases en lugar de la presencia de variables aleatorias, y la noción de conjuntos borrosos es completamente de naturaleza no estadística. Los conjuntos borrosos y los sistemas construidos a partir de ellos, han servido como una de las herramientas para resolver problemas en presencia de vaguedad; por ejemplo, cuando es necesario usar conceptos vagos como Alto, Viejo, etc.

Un sistema de inferencia borroso (SIB) es un sistema computacional basado en los conceptos de la teoría de conjuntos borrosos, reglas *If-Then* borrosas y razonamiento borroso. Se conocen por diversos nombres como Sistemas basados en reglas borrosos, Sistemas expertos borrosos, entre otros.

El razonamiento borroso (RB) o *modus ponens* generalizado, es un procedimiento de inferencia que deriva conclusiones a partir de un conjunto de reglas borrosas y hechos conocidos. Las entradas y salidas pueden ser valores duros o borrosos. Cuando la salida es borrosa y se necesita

el valor duro, se emplea un método de defuzificación (*defuzzyfication*) que determina el valor duro que mejor representa un conjunto borroso.

Es importante resaltar una característica de los SIB que lo distinguen de los Sistemas Basados en Reglas: el resultado de la inferencia se obtiene de aplicar numerosas reglas (usualmente todas las de la base) y conciliar las inferencias parciales de estas. La forma en que se concilian o integran todas estas inferencias parciales para producir un valor se denomina procedimiento de agregación. El producto de la agregación es el que posiblemente sea necesario defuzificar.

Seguidamente se presentan tres tipos de SIB, que se diferencian en la forma del consecuente de sus reglas borrosas y por eso varían los procedimientos de agregación y defuzificación [JYH98].

Modelo borroso de Mamdani.

En este modelo el antecedente y el consecuente de las reglas son conjuntos borrosos; la regla puede tener múltiples antecedentes. El método de inferencia de este modelo se basa en el empleo de un esquema de composición para producir el conjunto borroso final. Este esquema de composición indica como acotar el conjunto borroso consecuente de cada regla a partir de los valores de sus antecedentes y luego como integrar (agregar) todos los conjuntos borrosos resultantes en uno solo. Se reconocen dos esquemas de combinación el *max-min* y el *max-prod*. En ambos se utiliza el operador *max* (unión) para realizar la agregación de los conjuntos borrosos. Difieren en la forma de hallar el acotamiento del conjunto borroso que resulta de cada regla, en el primero se usa el operador mínimo y en el segundo el operador producto.

Modelo de Sugeno.

Este modelo, conocido también como TSK (por sus autores Takagi, Sugeno y Kang) utiliza reglas donde los antecedentes son conjuntos borrosos y el consecuente es una función dura. Cuando la salida de la regla es un polinomio de primer grado, el SIB resultante se denomina Modelo borroso de Sugeno de primer grado; mientras que si este es una constante se denomina Modelo borroso de Sugeno de cero grado. En este modelo no es necesario un procedimiento de defuzificacion pues el resultado de cada regla es un valor duro y la agregación de ellos produce también un valor duro. El procedimiento de agregación consiste en el promedio pesado de los valores resultantes de cada regla. Para encontrar el peso que afectara el valor del consecuente de cada regla en la suma ponderada se puede usar el operador mínimo o el operador producto (cuando hay múltiples antecedentes).

Modelo borroso de Tsukamoto

En este modelo el consecuente de cada regla borrosa se representa por un conjunto borroso con una MF monótona. El resultado inferido de cada regla es definido como un valor duro inducido por el acotamiento que resulta del antecedente de la regla. Para hallar el valor que acota (como en el modelo de Mamdani) o el valor que sirve de peso (como en el modelo de Sugeno) se pueden usar igualmente los operadores mínimo o producto (cuando la regla tiene múltiples antecedentes). El valor resultante se proyecta sobre la curva que describe la FP del conjunto borroso que aparece como consecuente, y luego se determina el valor del universo que tiene ese grado de membresía; ese será el valor duro resultante de la regla. El procedimiento de agregación consiste, al igual que en el modelo de Sugeno, en el promedio pesado de esos valores:

1.2 Los Sistemas Híbridos

En la actualidad la tendencia es desarrollar Sistemas Híbridos (SH), considerados estos una etapa superior de los SBC, para obtener un producto que aprovecha las ventajas de cada uno y minimiza sus deficiencias [HED93] [LIE93].

En [COR00] se presentan tres posibles clasificaciones de las diferentes formas de integración que se puede hacer en sistemas inteligentes: abc, IRIS y de Medsker y Bailey. A continuación se detallarán cada una de estas.

Clasificación abc

Los niveles de integración definidos en el modelo, están clasificados en función de la complejidad con la que interaccionan los subsistemas que componen dicho modelo. Esta clasificación identifica tres niveles de integración: artificial, biológica y computacional. En los tres niveles la complejidad aumenta al pasar del nivel de inteligencia computacional al nivel de inteligencia artificial, y de éste al de inteligencia biológica. En cada nivel se muestra cómo las pequeñas unidades de conocimiento (por ejemplo neuronas) pueden utilizarse para implementar técnicas de reconocimiento de patrones y así crear sistemas inteligentes. Un sistema híbrido en el modelo abc, es una forma de extender la inteligencia computacional a través de la inteligencia artificial, para alcanzar el objetivo final de modelar la inteligencia biológica.

Clasificación IRIS

El Modelo IRIS (Integración de Razonamiento, Información y Servicio) permite la combinación de software, hardware y distintos tipos de sistemas inteligentes. El objetivo de esta metodología

es facilitar el diseño eficiente de sistemas inteligentes, creando así productos y servicios que satisfagan completamente las necesidades de los negocios para los que fueron concebidos. Este enfoque se basa en la idea de que un sistema híbrido requiere la integración de diferentes disciplinas científicas incluyendo la biología, la psicología cognitiva, la lingüística y la informática.

Clasificación de Medsker y Bailey

Medsker y Bailey en [MED92] definen cinco modos de integración que se muestran en la figura 1 tomada de [COR00], donde SS representa a los sistemas simbólicos y SC a los sistemas conexionistas.

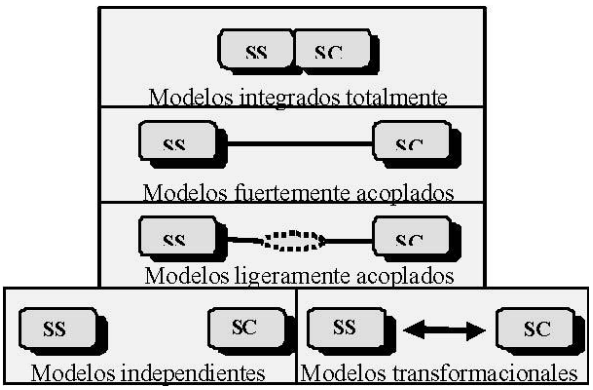


Figura 1: Modos de integración en sistemas de inteligencia artificial

A continuación, se explican brevemente cada una de estas formas de hibridación.

a) Modelos independientes

Este modelo de hibridación se caracteriza por la utilización de varios sistemas inteligentes de forma paralela sin interacción entre ellos. Se utiliza en situaciones donde se requiere la comparación de la eficacia y la eficiencia con la que dos sistemas de IA diferentes resuelven el mismo problema, verificar una solución dada por un sistema inteligente con otra proporcionada por un mecanismo diferente, o construcción de prototipos de forma rápida en paralelo con la intención de crear una aplicación sólida.

El uso de dos técnicas en paralelo para resolver un mismo problema provoca redundancia, pero a la vez el desarrollar una técnica después de otra es una buena forma de validar la primera. La utilización de dos técnicas en paralelo es una forma de integración muy débil que en ciertas ocasiones simplifica la tarea de implementación del sistema, especialmente si la aplicación se desarrolla por dos o más grupos de trabajo diferentes [COR00].

b) Modelos transformacionales

En este caso se empieza a resolver un problema utilizando un modelo inteligente y se termina empleando otro. Un ejemplo de este tipo de modelo se da en la situación en la que inicialmente se piensa que la mejor solución para resolver un problema es utilizar una RNA, y durante el transcurso de la implementación se decide que un sistema experto puede proporcionar una solución más adecuada si absorbe ciertas características de ésta.

Aunque este modelo permite que el desarrollo del sistema final se realice con la técnica más apropiada, y una implementación más rápida que el anterior; transformar un sistema experto en una RNA y viceversa automáticamente es difícil de implementar, si se requieren grandes transformaciones es muy probable que haya que empezar de nuevo y el sistema final tiene las limitaciones del modelo seleccionado [COR00].

c) Modelos ligeramente acoplados

Es la primera forma de auténtica integración de los sistemas inteligentes. En este caso una aplicación consta de varios sistemas inteligentes diferentes, que se comunican entre sí por medio de ficheros de datos. La integración se puede lograr de diferentes formas [COR00]:

- Preprocesador: una RNA se puede utilizar como un procesador de datos de entrada que realice funciones de modificación, selección, etc., de los datos antes de introducirlos a un SE.
- Postprocesador: los resultados obtenidos por un SE se pueden enviar a una RNA, por medio de un fichero, la RNA a partir de estos se encarga de hacer predicciones, análisis de datos, seguimiento de resultados, etc.
- Coprocesador: requiere la comunicación bilateral entre los dos (o más) sistemas que forman parte del híbrido, permitiendo un comportamiento de cooperación entre ellos.
- Interfaz de Usuario: una RNA se puede utilizar para facilitar la interacción entre un usuario y un SE.

Estos modelos son más fáciles de desarrollar que aquellos que tienen una integración más fuerte, como los que se verán a continuación. Sus mayores limitaciones son que el tiempo de operación es elevado (los protocolos de comunicación por medio de ficheros tienen asociado un alto coste de comunicación), y el desarrollo de sistemas inteligentes separados con posibilidades de comunicación entre ellos, conlleva una duplicación de esfuerzos.

d) Modelos fuertemente acoplados

Este modelo se diferencia del anterior en que la información entre los sistemas inteligentes se comparte por medio de estructuras residentes en la memoria, y no por medio de ficheros de datos externos. Esto incrementa la capacidad de interacción entre sistemas y aventaja la eficacia del modelo visto anteriormente, y también se puede utilizar en los cuatro submodelos presentados [COR00].

e) Modelos totalmente integrados

El modelo de integración total se caracteriza porque los distintos sistemas inteligentes que lo forman, comparten las estructuras de datos y la representación de los mismos. La comunicación entre los diferentes submódulos se efectúa utilizando la naturaleza dual (conexionista y simbólica) de las estructuras de datos. El razonamiento se realiza en la mayoría de los casos utilizando un modelo cooperativo, o mediante el uso de un componente que hace de controlador y optimiza el híbrido.

Las ventajas de este modelo son: el producto final es robusto y sólido, mejora de las capacidades de resolución de problemas; durante el desarrollo del sistema no existe redundancia; y mejora la capacidad de adaptación generalización, tolerancia al ruido, justificación y deducción lógica con respecto a sistemas no integrados. Las limitaciones del modelo son relativas al aumento en la complejidad del modelo pues se dificulta el desarrollo de herramientas que elaboren estos híbridos de forma automática; mientras que la validación, verificación y mantenimiento de estos sistemas requiere más trabajo.

Los sistemas expertos conexionistas es la variedad más representativa del modelo totalmente integrado [COR00]. Estos SH incorporan varios módulos neuronales y simbólicos que interactúan para resolver problemas eficientemente [SHI93], y se han nombrado de diferentes formas [CAU91] [SIM95] [TOW94]. La complementariedad de estos enfoques se fundamenta en el hecho de que dos módulos diferentes permiten modelar las habilidades cognoscitivas, el conexionista se usa para modelar el aspecto asociativo y el simbólico el aspecto lógico [GIA92]. Por otro lado, las RNA facilitan el trabajo con información incompleta y brindan poderosos algoritmos de aprendizaje que crean la base de conocimiento [SHI93] [CAU94] [SIM95] [TOW94], mientras que el enfoque simbólico favorece la representación explícita del conocimiento que hace posible la explicación.

En un SBC es conveniente dar una explicación al usuario del por qué la solución encontrada, dada su naturaleza heurística para resolver un problema. La forma más común de explicación es la de los sistemas basados en reglas que con esta finalidad emplean el razonamiento retrospectivo o razonamiento inverso [GON93]. Otra forma de explicación es la denominada Justificación, en la cual se presenta la información que fue relevante para obtener un resultado. En los SBCA se argumenta una solución mediante los casos que son relevantes al nuevo problema, y constituyen un ejemplo de este tipo de explicación.

Por ejemplo, en [BEL02] se presenta un sistema experto para el diagnóstico y tratamiento del embarazo Ectópico, que combina el RBC con reglas de producción. Cuando la similitud máxima del problema que se presenta con los casos almacenados en la bases de casos no es satisfactoria; el sistema resuelve el problema utilizando las reglas de producción, y a partir de este momento el problema resuelto pude ser incorporado a la base de casos.

Una especial relevancia y gran aplicación han tenido los sistemas expertos conexionistas que combinan las RBC y RNA. A continuación nos referiremos brevemente a las características de ambos modelos que favorecen este tipo de hibridación.

En las RNA es muy difícil explicar la forma de razonamiento y justificar sus resultados, sin embargo, sus capacidades de generalización y adaptación son muy interesantes y se pueden utilizar para resolver determinados problemas. Esto hace que no sean recomendables en situaciones que requieran la justificación del razonamiento. En la solución de problemas, las RNA se han destacado por su capacidad de generalización del conocimiento experto, y la propiedad de distribuir este entre los diferentes enlaces que la componen.

Como se mencionó anteriormente, la capacidad de generalización es una excelente característica de las RNA, pero en algunas ocasiones es conveniente mantener información relativa a casos concretos, y ésta es una capacidad natural de los sistemas SBCA. En los SBCA se pueden utilizar estas ventajas de las RNA para aprender a identificar la distancia más corta entre varios casos (y a partir de este criterio seleccionar los casos almacenados más similares al problema); e incluso, una RNA puede hacer que a un SBCA le resulte relativamente fácil generar nuevas soluciones, generalizar a partir de casos conocidos y adaptarlos a situaciones presentes [COR00].

Las RNA y los SBCA son técnicas complementarias: las primeras son más aconsejables en problemas de tipo numérico, mientras que los SBCA son más aconsejables en problemas

relacionados con el conocimiento simbólico. Aunque el conocimiento simbólico se puede transformar a una representación numérica y viceversa, hay que considerar que la transformación supone un riesgo de pérdida de información y precisión, por cuanto la combinación de estos enfoques supondría la eliminación de una transformación de este tipo, y por consiguiente un aumento en la precisión de los resultados [COR00].

Los SB y los que se desarrollan utilizando las RNA tienen en común la habilidad para mejorar la inteligencia de los sistemas que trabajan con incertidumbre, imprecisión, y en ambientes ruidosos. Además ambos han mostrado su habilidad de modelación en procesos complejos no lineales a un grado arbitrario de precisión. Entre la lógica borrosa y las RNA puede establecerse una relación bidireccional, ya que es posible, utilizar redes para optimizar ciertos parámetros de los sistemas borrosos, pero también se puede aplicar la lógica borrosa para modelar un nuevo tipo de neurona especializada en el procesamiento de información de este tipo [ZUR92] [HIL95].

Utilizando los llamados Conjuntos Borrosos el humano trata la información inexacta, mientras que las RNA son modelos de la arquitectura física y funcionamiento del cerebro de los humanos. En particular, estos han sido campos donde ha crecido el interés de los investigadores de varias áreas. Aunque las inspiraciones fundamentales de estos dos campos son bastante diferentes, hay muchos puntos en común que demuestran sus similitudes [CHI96].

Las diferencias radican en que los SB son estimadores numéricos estructurados. Ellos parten de vistas bastante formalizadas de la estructura de categorías encontradas en el mundo real, y entonces articular reglas borrosas para producir comportamientos complejos no lineales. Por otro lado las RNA son sistemas dinámicos entrenables, los cuales aprenden, tienen tolerancia al ruido, y habilidades generales de crecer fuera de su estructura conexionista y representación distribuida de los datos. En particular las RNA tienen un gran número de elementos de procesamiento altamente conectados, los cuales demuestran la habilidad para aprender y generalizar a partir de ejemplos de entrenamiento, y así simples elementos de procesamiento también colectivamente producen comportamientos no lineales complejos. Otra variante de combinación son los llamados sistemas neuro-borrosos. La idea de estos sistemas es encontrar los parámetros de un SB mediante el significado de los métodos de aprendizaje obtenidos desde una RNA. Los parámetros del SB pueden ser la cantidad de reglas de la base de reglas, así como los parámetros de las FP, entre otros. Los métodos de aprendizaje se refieren, por su parte, a los métodos que permiten reajustar esos parámetros que fueron previamente obtenidos en el SB, hasta lograr las reglas necesarias y el correcto funcionamiento de las FP a utilizar en la clasificación. Una forma común

para aplicar un algoritmo de aprendizaje a un SB está en representarlo en una arquitectura especial de una RNA. Entonces, un algoritmo de aprendizaje se usa para entrenar el sistema.

1.3 Aplicaciones de los Sistemas Híbridos

Los sistemas que combinan SB y RNA se denominan sistemas Neuro-Borrosos, lo cuales son frecuentemente representados como RNA multicapas dirigidas hacia delante. Algunas aproximaciones se abstienen de representar un sistema fuzzy en una arquitectura de red. Ellos solamente aplican un procedimiento de aprendizaje para los parámetros del sistema fuzzy, o explícitamente usan una RNA para determinarlo. Algunos ejemplos de estos sistemas son los siguientes:

- En [JYH98] se presenta el ANFIS, que es un sistema híbrido neuronal borroso que implementa un SB tipo Sugeno como una arquitectura de RNA hacia delante de cinco capas. La base de reglas debe conocerse previamente, ANFIS solo determina las FP de los antecedentes y parámetros lineales de los consecuentes de la regla, aplicando procedimientos estándares de aprendizaje de las RNA como el algoritmo de propagación de los errores hacia atrás (*Backpropagation*). El aprendizaje tiene dos fases, en la primera se estiman los parámetros óptimos del consecuente, asumiendo fijos los parámetros de los antecedentes; mientras que en la segunda fase los patrones se propagan nuevamente con el *Backpropagation* se modifican los parámetros de las FP, asumiendo fijos los consecuentes.

- El sistema NEFCLASS [JYH98] es un algoritmo que detecta, primeramente, todos los antecedentes de la regla que cubre algunos datos de entrenamiento y crea una lista de antecedentes. Al principio esta lista está vacía o contiene antecedentes de una regla dada como conocimiento previo. Cada vez que un patrón de entrenamiento es entrado y no es cubierto por un antecedente de la lista, un nuevo antecedente es creado y almacenado. Después, el algoritmo selecciona un consecuente apropiado para cada antecedente y crea una lista base de reglas candidatas. Para cada regla es calculada una medida de su funcionamiento, indicando su precisión (no ambigüedad).

- El sistema NEFPROX [JYH98] actúa similar al algoritmo anterior, pero para determinar los conjuntos borrosos y consecuentes, éste calcula un promedio pesado de la salida de los valores objetivos para todos los patrones que la función de membresía no es cero con un antecedente descubierto en los datos. El algoritmo de aprendizaje del NEXPROX necesita recorrer dos veces

el conjunto de entrenamiento, una para obtener los antecedentes de las reglas y otra para calcular la media y varianza para todos los antecedentes, requiere después, seleccionar los consecuentes para todos los antecedentes y completar los cálculos estadísticos.

- En [JYH98] se presenta el ANFIS, que es un sistema híbrido neuronal borroso que implementa un SB tipo Sugeno como una arquitectura de RNA hacia delante de cinco capas. La base de reglas debe conocerse previamente, ANFIS solo determina las FP de los antecedentes y parámetros lineales de los consecuentes de la regla, aplicando procedimientos estándares de aprendizaje de las RNA como el algoritmo de propagación de los errores hacia atrás (*Backpropagation*). El aprendizaje tiene dos fases, en la primera se estiman los parámetros óptimos del consecuente, asumiendo fijos los parámetros de los antecedentes; mientras que en la segunda fase los patrones se propagan nuevamente con el *Backpropagation* se modifican los parámetros de las FP, asumiendo fijos los consecuentes.

- En [HAT00] la generación de neuroreglas es un buen ejemplo de la integración entre RNA y la lógica difusa. Una neuroregla se considera como una unidad de la red adaline, donde los pesos representan factores de significación. Cada factor representa cuán significativa es la condición asociada en el dibujo de la conclusión. Se dispara una regla cuando la salida correspondiente del adaline llega a activarse. Modelar reglas simbólicas borrosas tradicionales como neuroreglas reduce el tamaño de la base de reglas y aumenta la eficacia de las inferencias.

- Castellano y Fanelli [CAS00] propusieron una RNA para construir y optimizar modelos borrosos. La red se puede mirar como un sistema borroso adaptativo con la capacidad de aprender reglas borrosas a partir de los datos, y a la vez como una arquitectura conexionista provista de significado lingüístico. Las reglas borrosas son extraídas de ejemplos del entrenamiento por un esquema de aprendizaje híbrido compuesto de dos fases: la fase de la generación de datos usando un aprendizaje competitivo modificado, y la fase que ajusta los parámetros usando aprendizaje del gradiente descendente. Esto permite la definición simultánea de la estructura y los parámetros de la base de reglas borrosas. Después de aprender, la red codifica en su topología los parámetros de diseño esenciales de un sistema de inferencia borroso.

Otra variante de SH es combinar las RNA y el RBC, donde la forma más común es aquella en la que los casos forman parte de una RNA. De esta manera, el RBC puede usar satisfactoriamente los casos en la etapa de recuperación y durante la indexación. Algunos ejemplos que se resumen en [COR00] son los siguientes:

- Thrift (1989) utiliza una RNA que aprende utilizando técnicas de retropropagación (*backpropagation*) para filtrar casos en la etapa de recuperación. La red selecciona los casos más relevantes de la memoria del SBCA.

-PATDEX/2 (Richter y Weiss, 1991) es un híbrido donde ambos mecanismos de razonamiento mantienen una relación completamente diferente a la que tienen los sistemas mencionados con anterioridad. PATDEX/2 es un sistema de diagnóstico de fallos basado en la forma de razonar de los CBR. Los casos son vectores de síntomas y el diagnóstico asociado a los mismos. La RNA utiliza un mecanismo de aprendizaje competitivo, y es el centro del mecanismo de recuperación de casos. El mecanismo de aprendizaje competitivo se basa en una matriz que asocia la relevancia de cada síntoma con cada diagnóstico posible. Los pesos de la matriz son modificados por una RNA en función del éxito de cada diagnóstico.

-Agre y Koprinska (1996) muestran un híbrido muy diferente a los vistos anteriormente. Su sistema combina un CBR con una RNA basada en conocimiento (*Knowledge-Based ANN*). Estas redes se caracterizan porque hacen uso del conocimiento del problema que quieren resolver para definir su estructura inicial. En este caso el CBR se utiliza solamente para corregir las soluciones erróneas de la RNA basada en conocimiento. Las correcciones son recuperadas por el CBR de entre los casos que se utilizaron para entrenar la RNA. Agre y Koprinska han probado que el híbrido mejora substancialmente la eficacia de la red.

- En el campo de los diagnósticos médicos, Reategui et al. (1996) han utilizado una RNA integrada en un CBR. El objetivo de la red es crear hipótesis y guiar el mecanismo de recuperación del CBR, que se encarga de buscar aquellos casos que coinciden con la hipótesis generada. El modelo ha sido creado para diagnosticar enfermedades coronarias congénitas. El sistema híbrido es capaz de resolver problemas que una red no puede solucionar por sí misma.

- Corchado y otros autores presentan un SH que combina RBC y RNA para predecir temperaturas oceanográficas de forma más acertada de lo que lo harían cualquiera de los sistemas de razonamiento utilizados en el híbrido por sí mismos. En uno de sus sistemas híbridos (Corchado et al., 1997), ambas técnicas se complementan entre sí y comparten la misma estructura de datos. Este sistema inicial, fue mejorado para realizar predicciones oceanográficas en tiempo real (Corchado, 1999). En este caso se utiliza una RNA en la fase de adaptación del CBR. La RNA se entrena con los casos recuperados y se utiliza para realizar una primera predicción. Este es un CBR con fuertes connotaciones sintácticas.

- Marinilli et al. (1999), presentan un híbrido CBR/ANN como sistema de filtrado de documentos de texto y HTML provenientes de la Web, basado en las preferencias y características del usuario final. Este sistema interactivo, emplea dos perceptrones multicapa encargados de clasificar el perfil del usuario y la categoría a la que pertenece el documento, y se utilizan para indexar casos en la memoria del CBR y construir las librerías de estereotipos y categorías de documentos.

- En [SAN01] se propone un modelo como vía para integrar el concepto de computación neurodifusa y RBC para el diseño de un sistema de toma de decisiones eficiente. Para este propósito se usa una red multicapas con neuronas difusas AND en la capa oculta y neuronas difusas OR como neuronas de salida. Los conjuntos lingüísticos difusos son considerados en la capa de entrada. Para manipular la incertidumbre y la vaguedad, los casos son descritos como reglas difusas IF-THEN. Se analizan además las relaciones entre los factores de peso en el antecedente de la regla, los factores de incertidumbre y los parámetros de la red.

- En [MAL01] emplean un Sistema Híbrido CBR-ANN que tiene como objetivo la identificación de comunidades de uso en el contexto de un grupo organizado de personas. La aplicación consiste en la recolección de ficheros marcados por los usuarios para identificar cuáles grupos de usuarios tienen los mismos intereses. Está estructurada como un conjunto de agentes asistentes, y un agente central que administra la organización de los usuarios. Cada agente emplea un clasificador CBR para aprender la estrategia de clasificación de ficheros de cada usuario. La similitud entre dos ficheros marcados se define mediante una función de agregación de dos funciones simples de similitud.

Esto son algunos ejemplos que muestran lo que se afirma en [SOV01]. La tendencia natural de la metodología de los RBC actuales es la integración de los mismos con otros sistemas expertos. Como se ha mostrado, esta integración puedes ser lograda incluso a través de la división de tareas entre el sistema RBC y la RNA, o con el diseño de una arquitectura inteligente combinando características de ambas tecnologías.

Capítulo 2: Componente Basada en casos de un Modelo Híbrido RNA-RBC

En varias aplicaciones [BEL02] del modelo híbrido presentado en [GAR96], la selección de los valores representativos de un rasgo lineal (continuo o discreto ordinal) se obtiene definiendo intervalos a partir del criterio de los expertos. Ideas preliminares de su extensión, utilizando los conjuntos borrosos, se presentan en [GAR00]. Aunque su aplicación en [ROD03], mostró resultados satisfactorios, se requiere de una validación extensiva del mismo a los efectos de estudiar los tipos de problemas donde es más recomendable su empleo.

Además en [ROD02] se aplica el modelo original al pronóstico de malformaciones cardiovasculares en recién nacidos, donde el rasgo objetivo se refiere a los tipos de malformaciones, y es usual encontrar en la BC ejemplos donde este rasgo toma múltiples valores. El modelo de RNA inicialmente implementado no permite lo anterior, y en tal sentido se incluyó el modelo IAC [MCC89]. Este modelo, aunque resuelve el problema planteado tiene la desventaja de requerir que en el tiempo de entrenamiento se estimen varios parámetros que utiliza la red en etapa de explotación.

En este capítulo se describe la herramienta que implementa el módulo basado en casos del modelo híbrido referenciado. Primeramente se describe en general el modelo a implementar, haciendo énfasis en las modificaciones realizadas a la componente basada en casos del modelo original.

2.1 Modelo Híbrido RNA-RBC utilizando conjuntos borrosos

El modelo híbrido que se presenta en [GAR96] combina las RNA y el RBC en la solución de problemas, donde ante la descripción de un problema a resolver (patrón P) la RNA completa el patrón (valoriza los rasgos definidos como objetivos) y la componente RBC se emplea para justificar la solución anterior en el contexto de los k casos más similares al problema P. De esta forma se aprovecha las ventajas que las RNA tienen en cuanto a la capacidad de aprendizaje y se elimina su limitación de no permitir explicar la solución encontrada.

Los modelos de RNA implementados son del tipo asociativo, es decir, no requieren una definición previa al entrenamiento de cuáles rasgos se consideran predictores y cuáles objetivos. En su topología existe un grupo de neuronas asociado a cada rasgo, donde se coloca una neurona por cada valor representativo del rasgo. Existen enlaces entre cada par de neuronas asociando un

peso ($w_{i,j}$) que indica en que medida los valores que éstas representan están relacionados en la BC.

La segunda componente de este modelo utiliza el RBC. Para recuperar los casos más similares a un problema *P* se utiliza una función de similitud [RUI93], la cual en el criterio de comparación a nivel de rasgo emplea los pesos de la RNA, obteniéndose de esta forma funciones de comparación generales e independientes del dominio de cada rasgo [GAR96].

En el nuevo modelo [GAR00] los valores representativos para un rasgo lineal se corresponden con los términos lingüísticos definidos cuando el rasgo fue modelado como variable lingüística. Si para un rasgo de este tipo se definen intervalos o el rasgo es simbólico se procede de igual forma que en el modelo original para conformar la topología de la RNA. Es por ello que en este nuevo modelo se requiere añadir a la RNA una capa de preprocesamiento, donde cada neurona tiene definida una función *f*, donde si *v* es el valor dado a un atributo *x* que tiene asociado un valor representativo α_i, este activará la neurona asociada con α_i en la RNA en la medida del resultado de la función $f_{\alpha i}(v)$ que se define a continuación.

$$f_{\alpha_i}(v) = 1 \qquad si \;\; \alpha_i \subseteq v \,, \;\; \forall \alpha_i \in T_x \tag{1}$$

$$f_{\alpha_i}(v) = \begin{cases} 1 & si \quad Inf_{\alpha_i} \le v \lhd Sup_{\alpha_i}, \forall \alpha_i \in T_x \\ 0 & en \quad otro \quad caso \end{cases} \tag{2}$$

$$f_{\alpha_i}(v) = \begin{cases} \mu_{\alpha_i}(v) & si \quad \mu_{\alpha_i}(v) = \max \mu_{\alpha_i}(v), \forall \alpha_i \in T_x \\ 0 & en \quad otro \quad caso \end{cases} \tag{3}$$

$$f_{\alpha_i}(v) = \begin{cases} \mu_{\alpha_i}(v) & si \quad \mu_{\alpha_i}(v) \ge \alpha_0, \forall \alpha_i \in T_x \quad y \quad \alpha_0 \in [0,1] \\ 0 & en \quad otro \quad caso \end{cases} \tag{4}$$

$$f_{\alpha_i}(v) = \mu_{\alpha_i}(v) \tag{5}$$

donde:

T_x : conjunto de valores representativos del atributo c

$\mu_{\alpha_i}(v)$: pertenencia de *v* al valor representativo α_i (término lingüístico) según la función μ

La primera expresión se aplica cuando el atributo c es nominal (o simbólico) y la segunda para atributos lineales discretizados (*Inf* y *Sup* se refieren a los extremos del intervalo α_i). Cuando se usan conjuntos difusos para modelar el atributo c, cualquiera de las tres últimas expresiones se pueden utilizar.

2.1.1 Módulo basado en casos para justificar

La explicación basada en casos consiste en justificar la solución dada al problema por la RNA presentando casos similares, almacenados en la BC, al problema resuelto. Esta recuperación se realiza utilizando una medida de similaridad que compara el problema resuelto (P) con cada caso R de la BC; tomando valores en el intervalo [0,1], donde uno significa máxima similitud. La expresión es la siguiente:

$$\beta(P,R) = \frac{1}{|Cp|} * \sum_{c \in Cp} \delta(P.c, R.c) \tag{6}$$

$$\delta(P.c, R.c) = \begin{cases} 1 & si \quad P.c = R.c \\ W_c(P) * 0.5 & si \quad R.c = ? \\ 0.5 & si \quad P.c = ? \\ W_c(P) * S_c(P.c, R.c) & en\ otro\ caso \end{cases} \tag{7}$$

Donde:

P.c y R.c: conjunto de valores asignados al rasgo c en el problema y en el caso respectivamente.

$W_c(P)$: importancia del rasgo c en el problema P.

S_c: equivalencia de los valores asignados al rasgo c en el problema y en el caso.

La función de comparación (expresión 7) alcanza valor 1 cuando el valor que toma el rasgo c en el caso y en problema a resolver coinciden. Si no apareciera el valor de este rasgo en el problema a resolver no se considera total igualdad ni total diferencia, sino el valor 0.5 como se propone en [RUI93]. Pero nótese que si este se desconoce solamente en el caso con el cual se está comparando, este valor se afecta por la importancia del valor del rasgo que se compara en el problema (ver ecuación 8). En el resto de las situaciones el valor resultante de esta medida de similitud local se calcula teniendo en cuenta esta medida de importancia y la equivalencia del valor que toma este rasgo en los patrones a comparar.

$$W_c(P) = \frac{1}{|C_o|} * \sum_{t=1}^{|C_o|} \frac{\sum_{\forall \alpha_j \in To t} \sum_{\forall \alpha_i \in Tc} w_{\alpha_i, \alpha_j} * f_{\alpha_i}(P.c) * f_{\alpha_j}(P.o_t)}{\sum_{\forall \alpha_j \in To_t} \sum_{\forall \alpha_i \in Tc} f_{\alpha_i}(P.c) * f_{\alpha_j}(P.o_t)} \tag{8}$$

donde:

To_t: conjunto de valores representativos del t-ésimo rasgo objetivo.

$P.o_t$: valor del t-ésimo rasgo objetivo en el problema P.

$w_{\alpha i \alpha j}$: arco de peso entre las neuronas i y j correspondientes a los valores representativos α_i y α_j [1] respectivamente.

$f_i(x)$: preprocesamiento del valor *x* según la función *f* de la neurona que representa el valor *i*, según alguna de las expresiones del epígrafe anterior.

La equivalencia entre el valor dado al atributo *c* en el problema y en el caso se calcula mediante la siguiente expresión:

$$S_c(P.c, R.c) = \frac{1}{|C_o|} * \sum_{t=1}^{|C_o|} \left[1 - \left| \frac{\sum_{\forall \alpha_j \in To_k} \sum_{\forall \alpha_i \in Tc} w_{\alpha_i,\alpha_j} * f_{\alpha_i}(P.c) * f_{\alpha_j}(P.o_k)}{\sum_{\forall \alpha_j \in To_k} \sum_{\forall \alpha_i \in Tc} f_{\alpha_i}(P.c) * f_{\alpha_j}(P.o_k)} - \frac{\sum_{\forall \alpha_j \in To_k} \sum_{\forall \alpha_i \in Tc} w_{\alpha_i,\alpha_j} * f_{\alpha_i}(R.c) * f_{\alpha_j}(R.o_t)}{\sum_{\forall \alpha_j \in To_k} \sum_{\forall \alpha_i \in Tc} f_{\alpha_i}(R.c) * f_{\alpha_j}(R.o_t)} \right| \right] \quad (9)$$

donde:

$R.o_t$: valor del t-ésimo rasgo objetivo en el caso R.

Los demás términos en la expresión se explicaron anteriormente.

2.1.2 Módulo basado en casos para resolver

La capacidad de una RNA para generalizar puede producir mejores resultados que el CBR. Para ello la RNA debe ser entrenada con una significativa cantidad de datos representativa de todas las situaciones que se puedan presentar en el dominio de aplicación. En la aplicación a un problema real donde inicialmente no se tengan suficientes ejemplos el RBC pudiera ser más efectivo que una RNA. Luego, como resultado de ir incorporando problemas resueltos y apropiados a la BC, la RNA pudiera ser reentrenada como se plantea en [LEE02]. Además se pueden presentar problemas que por sus características, aunque la RNA haya logrado hacer generalizaciones, constituyen excepciones del dominio de aplicación cuya solución no se ajusta a la generalidad de los casos. En situaciones como esta el RBC podría lograr mejor desempeño respecto a la RNA.

[1] Ver [GAR00] para saber como calcular esta medida.

Teniendo en cuenta estas reflexiones surge la idea de extender el modelo original añadiendo una nueva funcionalidad, la de utilizar el RBC para resolver un problema. Para ello la componente basada en casos originalmente empleada se modifica manteniendo la generalidad de la FS definida, para proponer una solución al problema que se presenta en el contexto de los k casos de la BC más similares a éste. A partir de las expresiones 7 y 8 del módulo anterior, se hacen modificaciones que se muestran a continuación, obteniéndose así una función de semejanza modificada que denotaremos por *β'*.

$$W_c(P) = \frac{1}{|C_o|} * \sum_{t=1}^{|C_o|} \frac{\sum_{\forall \alpha_j \in To_t} \sum_{\forall \alpha_i \in Tc} w_{\alpha_i,\alpha_j} * f_{\alpha_i}(P.c)}{\sum_{\forall \alpha_j \in To_t} \sum_{\forall \alpha_i \in Tc} f_{\alpha_i}(P.c)} \tag{10}$$

$$S_c(P.c, R.c) = \frac{1}{|C_o|} * \sum_{t=1}^{|C_o|} \left[1 - \left| \frac{\sum_{\forall \alpha_j \in To_t} \sum_{\forall \alpha_i \in Tc} w_{\alpha_i,\alpha_j} * f_{\alpha_i}(P.c)}{\sum_{\forall \alpha_j \in To_t} \sum_{\forall \alpha_i \in Tc} f_{\alpha_i}(P.c)} - \frac{\sum_{\forall \alpha_j \in To_t} \sum_{\forall \alpha_i \in Tc} w_{\alpha_i,\alpha_j} * f_{\alpha_i}(R.c)}{\sum_{\forall \alpha_j \in To_t} \sum_{\forall \alpha_i \in Tc} f_{\alpha_i}(R.c)} \right| \right] \tag{11}$$

El significado de cada uno de los términos empleados es el mismo que en las expresiones anteriores. Las modificaciones realizadas siguen un criterio similar al utilizado por Stanfill y Waltz en la métrica VDM [STA86], la cual considera que el conjunto de soluciones posibles está formado por las clases definidas en el problema que se resuelve. Recuerde que esta métrica se emplea en SBCA para resolver solamente problemas de clasificación.

La expresión 10 calcula la importancia de un rasgo considerando la influencia que los valores dados a *P.c* tienen, según la BC, en los valores posibles para los rasgos en *Co*. Según la expresión 11 la similitud de dos patrones en cuanto a un rasgo *c* se estima considerando la proyección sobre la BC, en cuanto a los rasgos en *Co*, de los valores dados a este en el patrón *P* y el caso *R*.

Atendiendo a lo anterior, para obtener los *k* casos más similares se forma un espacio de soluciones posibles considerando el dominio (valores presentes en la BC) de todos los rasgos en *Co*. Para un problema de clasificación, este se formaría por el conjunto de clases definidas en el problema, al igual que en VDM. De esta misma forma se define el espacio de solución de problemas más generales de búsqueda asociativa de información, para los cuales la división del conjunto *M* de rasgos en predoctores y objetivos puede variar de un problema a otro que se quiera resolver.

La complejidad computacional de estos cálculos se simplifica considerablemente a partir de que la información de la BC está previamente codificada en los pesos de la RNA, a partir del entrenamiento de la misma, considerando la BC como conjunto de entrenamiento. Nótese que en estas expresiones la información requerida para hacer las comparaciones y el cálculo de la similaridad se toma de los pesos de la RNA, lo que evita el acceso reiterado a la BC.

Para resolver un problema primeramente se recuperan de la BC los *k*-casos más similares a éste utilizando la función de semejanza *β'*. A partir de estos, y haciendo una generalización para este modelo de la variante utilizada en [WET97] [WIL00] la solución del problema se obtiene utilizando la siguiente expresión:

$$P.o = ArgMax \text{Prob}^{k}_{c \in C'}(P, c) \tag{12}$$

$$\text{Prob}^{k}(P, c) = \frac{\sum_{i=1}^{k} \beta'(P, R^i) * \delta(R^i.o, c)}{\sum_{i=1}^{k} \beta'(P, R^i)} \tag{13}$$

Donde:

δ: es una medida de la coincidencia entre dos soluciones (expresión 18)

El conjunto *C'* se forma considerando todas las combinaciones posibles (sin repetición) de los valores presentes para los rasgos objetivos en los k casos más similares, es decir el cardinal de este conjunto sería:

$$\sum_{j=1}^{k} \left(\sum_{t=1}^{m_1{}^{j}} C_{m_1{}^{j}, t} \right) \times \ldots \times \left(\sum_{i=1}^{m_T{}^{j}} C^{k}_{m_T{}^{j}, t} \right) \tag{14}$$

Donde:

m_t^k : es la cantidad de valores presentes para el *t*-ésimo rasgo objetivo del *j*-ésimo caso más similar (*t= 1..T, donde T=/Co/)*

A partir de lo anterior se puede comprender que de los *k* casos más similares de la BC encontrados para un problema *P*, depende la calidad de la solución inferida para éste. Este valor varía de una BC a otra; y existe la teoría de que un número indicado para *k* debe ser 7±2, pensando en la forma que razona un humano.

El valor del *k* (ideal) para una BC se puede calcular considerando el LOOCE (*leave-one-out cross error*) [WIL00], según la expresión:

$$k_{óptimo} = \min_{k=1..15} LOOCE_k \qquad (15)$$

$$LOOCE_k = \frac{\sum_{i=0}^{|BC|} e_i^k}{|BC|}$$

Donde: e_i^k es una estimación del error cometido en resolver un problema a partir de los *k*-casos más similares.

La estimación de este error puede variar, ejemplos de algunas medidas se presentan en [WET97]. La variante utilizada se especifica en la expresión siguiente:

$$e_i^k = Dist\left(ArgMax \text{Prob}^k_{c \in C'}(R^i, c), R^i.o\right) \qquad (16)$$

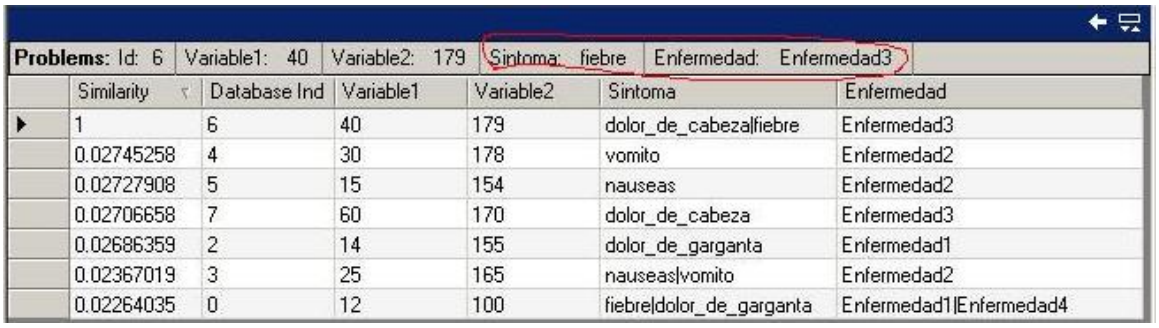
Donde: *Dist* es una medida de distancia (ver expresión 19 y 20)

Para ilustrar lo anterior considere un problema donde cada caso de la BC se describe por los siguientes atributos:

Atributo	Valores posibles
Edad	Intervalos: (8;14), (14;35), (35;60)
Estatura	Términos lingüísticos bajo, mediano y alto
Síntomas	fiebre, dolor_de_garganta, náuseas, vómito, dolor_de_cabeza
Enfermedad	Enfermedad1, Enfermedad2, Enfermedad3

Tabla 1. Descripción de los atributos

Considerando como Co={Síntomas, Enfermedad}, para dar solución al problema: (40, 179, ?, ?), los siete casos más similares recuperados de la BC (el k utilizado en este ejemplo no es el óptimo), y la solución que se propone por el módulo basado en casos se muestran en la figura 2. Nótese que la solución encontrada no aparece explícitamente como solución de los casos similares considerados, sino que forma parte del conjunto *C'* definido a partir de éstos mediante la expresión 14.



Problems: Id: 6 | Variable1: 40 | Variable2: 179 | Sintoma: fiebre | Enfermedad: Enfermedad3

Similarity	Database Ind	Variable1	Variable2	Sintoma	Enfermedad
1	6	40	179	dolor_de_cabeza|fiebre	Enfermedad3
0.02745258	4	30	178	vomito	Enfermedad2
0.02727908	5	15	154	nauseas	Enfermedad2
0.02706658	7	60	170	dolor_de_cabeza	Enfermedad3
0.02686359	2	14	155	dolor_de_garganta	Enfermedad1
0.02367019	3	25	165	nauseas|vomito	Enfermedad2
0.02264035	0	12	100	fiebre|dolor_de_garganta	Enfermedad1|Enfermedad4

Figura 2. Resultados del módulo RBC para resolver el problema (40, 179, ?, ?).

2.2 Herramienta para la componente basada en casos de un modelo híbrido RNA-RBC

La herramienta que se implementó brinda la posibilidad de ejecutar el modelo híbrido anteriormente explicado frente a una cierta base de casos. Se utilizó el lenguaje de programación Microsoft C#.NET por su gran comodidad en el manejo de diseños orientados a objetos y su alto nivel de abstracción de muchas de las funcionalidades con las que tienen que lidiar los programadores en otros lenguajes, tales como la liberación de la memoria utilizada y la administración de punteros.

Por tanto, se requiere que en la PC donde se ejecute esta herramienta se encuentre el .NET *Framework* versión 1.1 o superior. En el caso del sistema operativo Windows 2003 Server, ya estas bibliotecas vienen como parte de su *kernel*.

2.2.1 Implementación computacional

Para la implementación de las funciones de semejanza utilizadas en el módulo se tuvo en cuenta la modelación del problema mediante el enfoque orientado a objetos, diseñándose una jerarquía de clases, con el objetivo de estandarizar un prototipo de función de similitud; de forma tal que se facilite la extensión del módulo al empleo de nuevos tipos de funciones de similitud sin necesidad de hacer modificaciones al código fuente de la herramienta.

Partiendo de la determinación de las principales funcionalidades que debe brindar el modelo de RBC a validar, se diseñó la clase *SimilarityFunctionEngine.* La misma define todos los métodos que deben ser implementados para que se logre cumplir con los requerimientos de un CBR. Esta clase es abstracta debido a que su objetivo principal, como se decía anteriormente, es definir un prototipo. A continuación se describen los principales métodos definidos para esta clase.

GetIdealK() le permite al RBC hallar el k ideal para una base de casos, el mismo se apoya en el algoritmo LOOCE, el cual calcula el error que comete el sistema al clasificar cada instancia de la base de casos, quedando este fuera de las posibles soluciones, este procedimiento se hace para distintos valores de k, desde k=1 hasta k=15, *LOOCE()* le permite al usuario conocer el posible error que se comete para el valor de k actual. Este error se puede traducir como el dual de la distancia entre el problema resuelto y la solución hallada para el mismo, se recuerda que el problema a resolver en este algoritmo es cada uno de los casos de la base de casos.

GetDistance() retorna la distancia entre dos casos, esta es calculada utilizando las membresías de los rasgos objetivos. Existen distintas variantes para calcular esta distancia, una de las principales metas del programa es permitir la incorporación de nuevos métodos para este cálculo fácilmente, hasta el momento sólo fueron implementadas dos variantes: la distancia Euclidiana y la distancia de *Manhattan*, esta última es la recomendada debido a que es más flexible para distintos tipos de definiciones de rasgos; en el caso de la Euclidiana sólo es aconsejable cuando se trabaja con un modelo que no incluye rasgos borrosos.

GetWinner() selecciona uno de los k vecinos más cercanos recuperados de la base de casos, para esto escoge la solución de mayor probabilidad, que no es más que la solución más representativa de las k halladas.

RetrieveSimilarCases() implementa la función de semejanza para recuperar los k vecinos más similares al caso. Este método está sobrecargado con el objetivo de permitir la resolución de un lote de problemas o sólo uno de ellos. El resultado de este método -los casos recuperados- se almacena en una lista de *SimilarityPairs*, un objeto diseñado con el objetivo de almacenar únicamente el índice, en la base de casos, del caso recuperado y la similitud del mismo con el problema, debido a que guardar toda la información del caso podría ser costoso en cuestiones de memoria, además de que si el mismo ya existe en memoria no es necesario duplicarlo. Este método se apoya en el uso de dos clases más, *TraitsSimilarity* y *TraitImportance*, ambas también constituyen un estándar, debido a que su implementación varía de un rol a otro de dicha función de semejanza.

La primera de estas clases se encarga, principalmente, de hallar la similitud entre dos rasgos predictores. El método principal en esta clase es *GetTraitsSimilarity()*, el mismo se ocupa de hacer el cálculo anteriormente explicado.

En el caso de la clase *TraitImportance*, esta define los métodos para hallar la importancia de un rasgo predictor en el problema. Su principal método es *GetWC()* para el cálculo de la importancia de un rasgo.

Siguiendo esta idea, se implementaron dos roles distintos de RBC. El primer rol implementado es el de **justificador**, para esto se diseñó la clase *SimilarityFunctionToJustify*, la misma no incluye grandes cambios con respecto a la arquitectura antes explicada, debido a que este es el rol más sencillo que desempeña el RBC y fue a partir de este que se definieron las principales funcionalidades de un RBC. Para esta clase se diseñaron las clases *TraitsSimilarityToJustify* y *TraitImportanceToJustify*, las mismas cumplen las funciones de hallar la similitud entre dos rasgos y de calcular la importancia de un rasgo respectivamente, cuando el RBC está desempeñando el rol de justificador.

El segundo rol de RBC implementado en la herramienta es el de **solucionador**, dotando al RBC de la habilidad de procesar los problemas y darle solución a estos. Con este fin se diseñó la clase *TraitImportanceToSolve*, uno de los nuevos métodos que contiene esta clase es *GetClass()*, debido a que a la hora de resolver no solo se toman como solución los valores de los rasgos objetivos de los k vecinos recuperados, además de estas se generan todas las posibles soluciones presentes en cada uno de estos vecinos. Estas nuevas soluciones sólo aparecen cuando al menos uno de los rasgos objetivos puede tomar más de un valor a la vez, de esta manera se forman todas las posibles combinaciones sin repeticiones de los valores del rasgo, esto se hace para los valores de cada rasgo objetivo del caso y luego se combinan las mismas de forma tal que no quede ninguna solución presente en el caso fuera de análisis. Las nuevas soluciones sólo son insertadas si no aparecen en ninguno de los casos recuperados o si no fue generada con anterioridad por unos de los casos ya analizados. Para hallar la similitud entre rasgos y la importancia de un rasgo en este nuevo rol, se diseñaron las clases *TraitsSimilarityToSolve* y *TraitImportanceToSolve* (Ver anexo 2).

2.1.1 ¿Cómo utilizar la herramienta?

El propósito general de esta herramienta es la validación de un modelo híbrido CBR-ANN que emplea una función de similitud, la cual hace uso de los pesos provenientes de un modelo de RNA asociativo. Esta herramienta será utilizada por un especialista en IA que haga uso del modelo híbrido implementado y desee validar sus resultados.

Para probar el desempeño del modelo híbrido frente a una determinada base de casos, se requieren:

- La base de casos (ej. Iris.data).
- La matriz resultante de la salida del preprocesamiento de la BC utilizando un fichero de definición de rasgos (ej. IrisMatrix.txt).
- La matriz de pesos de esta BC generada por el modelo de RNA (ej. IrisWeights.txt).
- El fichero con los índices de los rasgos objetivos y los nombres de las neuronas preprocesadoras para cada rasgo (ej. IrisPools.txt).

Como la generación de los dos últimos ficheros por la RNA requiere de la creación de un fichero experto (*.nex) en la herramienta NeuroDeveloper, que indica la ubicación de los mismos, nuestra aplicación crea un proyecto tomando como entrada el fichero experto generado por la herramienta que tiene implementada la RNA.

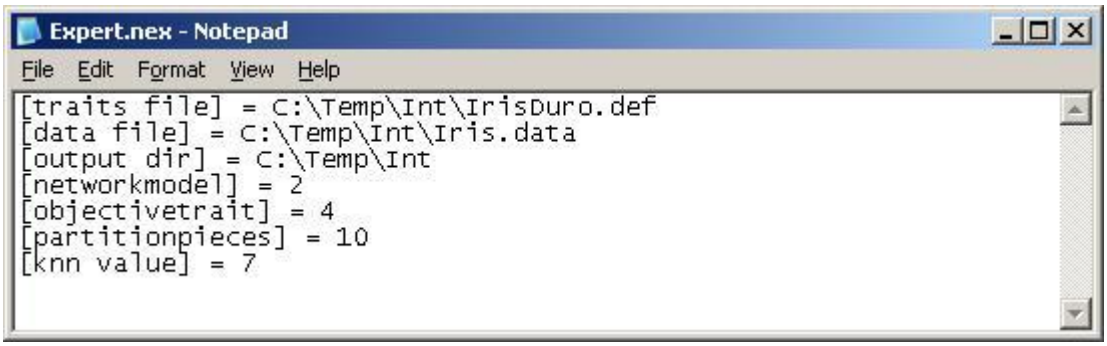


Figura 3. Fichero experto generado por la RNA.

La RNA genera una serie de ficheros; los principales que son usados para crear un nuevo proyecto en la herramienta, son generados hacia el directorio de salida especificado en el experto y llevan como prefijo el nombre del experto. Es por esto que, por ejemplo, el fichero de preprocesamiento para el ejemplo mostrado en la figura 3 se podrá encontrar en el directorio C:\Temp\Int y con el nombre de ExpertMatrix.txt, en el caso del fichero de pesos se sustituye *Matrix* por *Weights* o *Pools* en el caso del fichero con los grupos de neuronas de preprocesamiento de cada rasgo en la BC.

Figura 4. Creación de un nuevo proyecto.

Luego de creado un proyecto, figura 4, este puede ser validado (c*ross-validation*). Para que esto ocurra, la propiedad *partitionpieces* del experto debe ser mayor que uno; si esto se cumple, entonces con sólo activar esta opción, la herramienta buscará las particiones en el directorio de salida del experto, donde cada partición está compuesta por:

- Una muestra de aprendizaje, que lleva por nombre *MANatural*, precedida por el nombre del experto y como sufijo el número de la partición.
- La matriz de preprocesamiento de dicha muestra de aprendizaje con el nombre *MA* y manteniendo el prefijo y el sufijo de la muestra de aprendizaje.
- La muestra de control nombrada *MCNatural* y con el prefijo y el sufijo antes explicado.
- La matriz de preprocesamiento de esta muestra de control con el nombre de *MC* y manteniendo el prefijo y el sufijo de la muestra de control.

Nota: Todos estos ficheros tienen extensión TXT.

Para la validación se tomarán como parámetros de la función los especificados en el nuevo proyecto creado, si se desea validar el mismo para el rol de justificador o de resolvedor o utilizando un K y una distancia distinta, entonces esto se deberá especificar en la ventana de propiedades, figura 5. El proceso de validación genera un nuevo fichero hacia el directorio de salida, el mismo contiene la precisión de cada partición y lleva por nombre el del proyecto creado la herramienta más *Statistics*, si el proyecto no ha sido salvado aún o su nombre no se ha cambiado en la ventana de propiedades, entonces este fichero llevará por nombre untitledStatistics.txt.

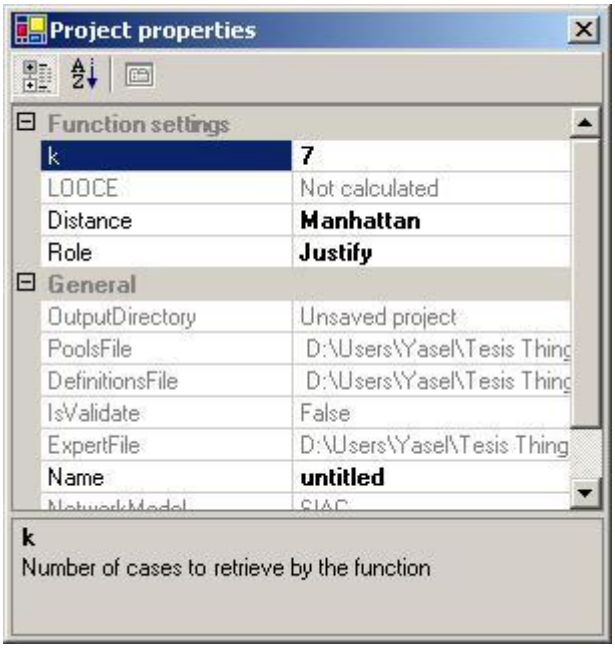


Figura 5. Ventana de propiedades.

Otras de las funcionalidades añadidas a la herramienta es el cálculo del k óptimo, de esta forma el usuario puede lograr un mejor desempeño de la misma. Además cuenta con la posibilidad de justificar un problema resuelto por la RNA o puede buscar sus propias soluciones para un problema, esto está en dependencia del rol que esté desempeñando el RBC en ese momento, figura 6.

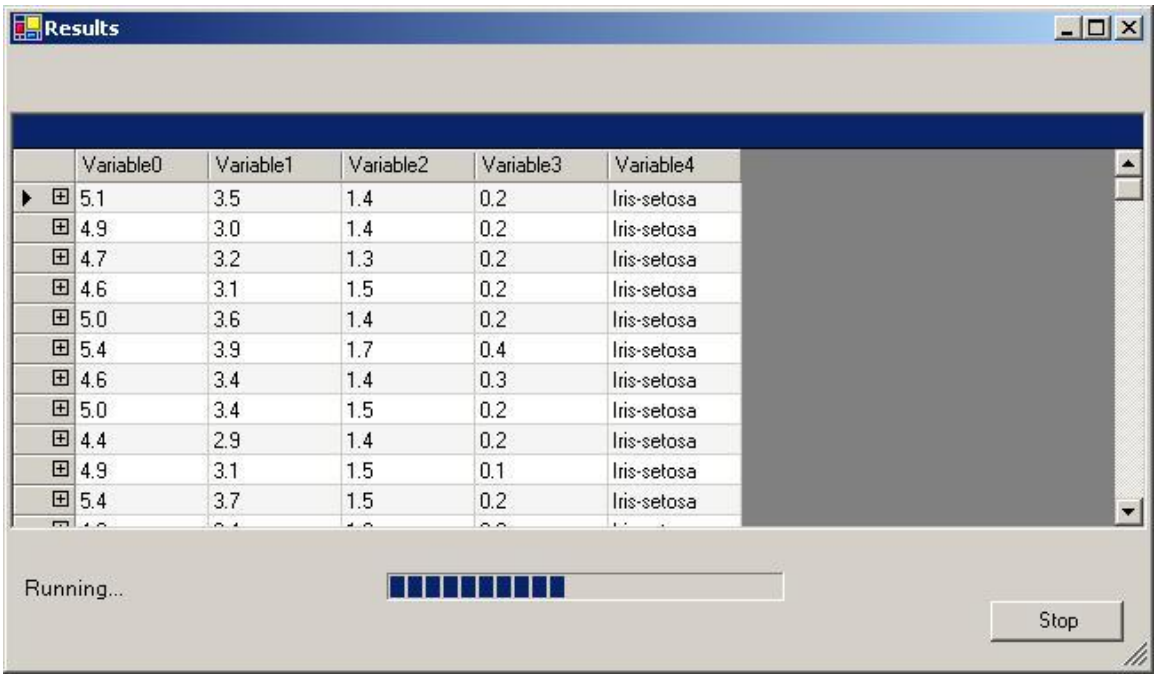


Figura 6. Ventana con los problemas introducidos.

Desde esta vista se puede pasar a otra con la solución particular de cada problema, figura 7, la primera fila mostrada en esta vista es la información del problema resuelto. Luego se muestra una tabla donde la primera columna muestra la similitud de cada caso recuperado con el problema, la segunda columna muestra el índice del caso en la base de casos y si el rol es de justificador, entonces se muestra al final una columna donde se marca al caso seleccionado como ganador de entre los k vecinos recuperados. Esta no tiene sentido cuando el RBC está desempeñando el rol de resolvedor debido a que en esta variante lo que se hace es asignarle al problema la nueva solución encontrada.

En la ventana de los problemas se muestra la precisión obtenida al justificar los problemas, y en la vista de las soluciones se muestra la distancia de la solución hallada a la obtenida por la RNA. En el caso que el rol sea resolvedor, estos datos sólo aparecerán si los problemas resueltos tenían alguna solución previa, en otro caso estos cálculos serán indeterminados.

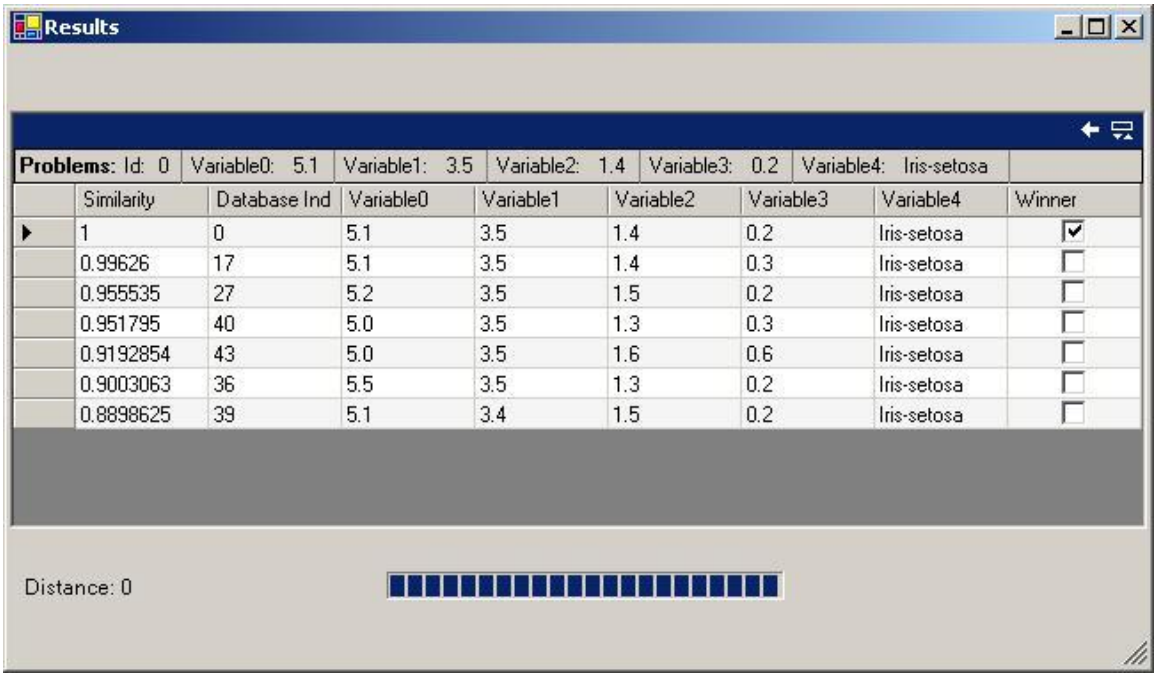


Figura 7. Vista con la justificación de un problema.

Seguidamente mostramos en la figura 8 el diagrama de componentes de la herramienta. El motor de inferencia del RBC depende para su funcionamiento de los datos contenidos en los ficheros de

base de casos, de pesos generados por la RNA y de la matriz de preprocesamiento de las entradas.

Como el RBC puede asumir el rol de resolvedor o el de justificador según el problema, las entradas y salidas del motor de inferencia del RBC estarán en función de dicho rol. Ya sea que se vaya a justificar o a resolver, el RBC recibe como entrada un lote de problemas y emite como salida la solución correspondiente al lote anterior, la cual contiene para cada problema del lote los k vecinos más cercanos así como el desempeño logrado por el RBC. Otra posible entrada al sistema sería un conjunto de particiones de la base de casos empleada que debe haber sido generado previamente. El objetivo de estas particiones es lograr una validación cruzada para dar una idea del desempeño del sistema, devolviendo como salida los resultados de la validación.

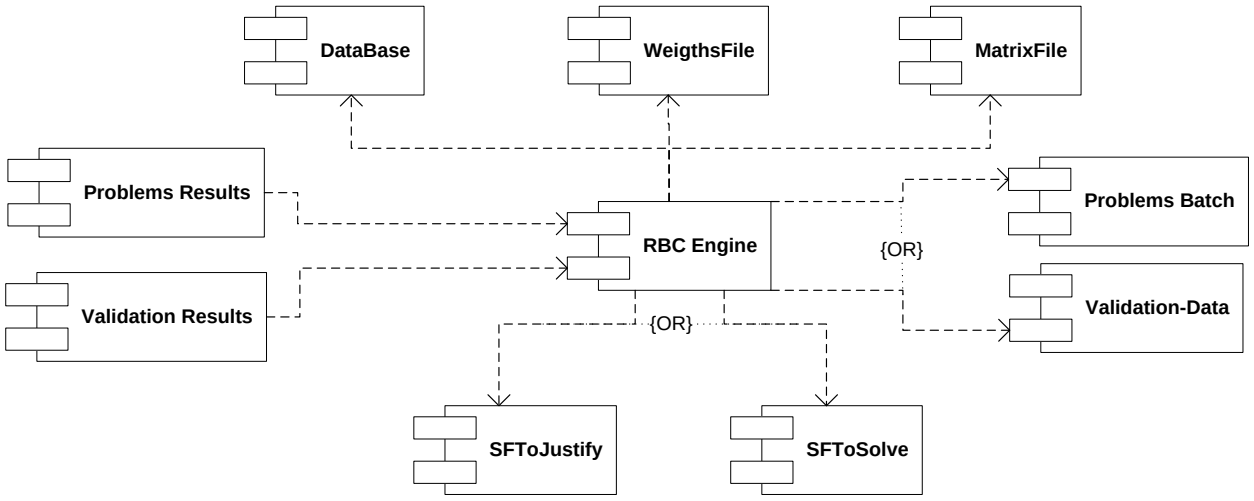


Figura 8. Diagrama de componentes.

El diagrama de estados representado en la figura 9 muestra las transiciones de estado por las que puede transitar la herramienta en su funcionamiento. Desde la ventana principal se invocan funcionalidades como realizar validación cruzada, mostrar las propiedades de un proyecto, retornar el K ideal, etc. las cuales transfieren el control del sistema hacia los módulos que las ejecutan, devolviéndose nuevamente el control hacia la ventana principal una vez finalizadas las

operaciones.

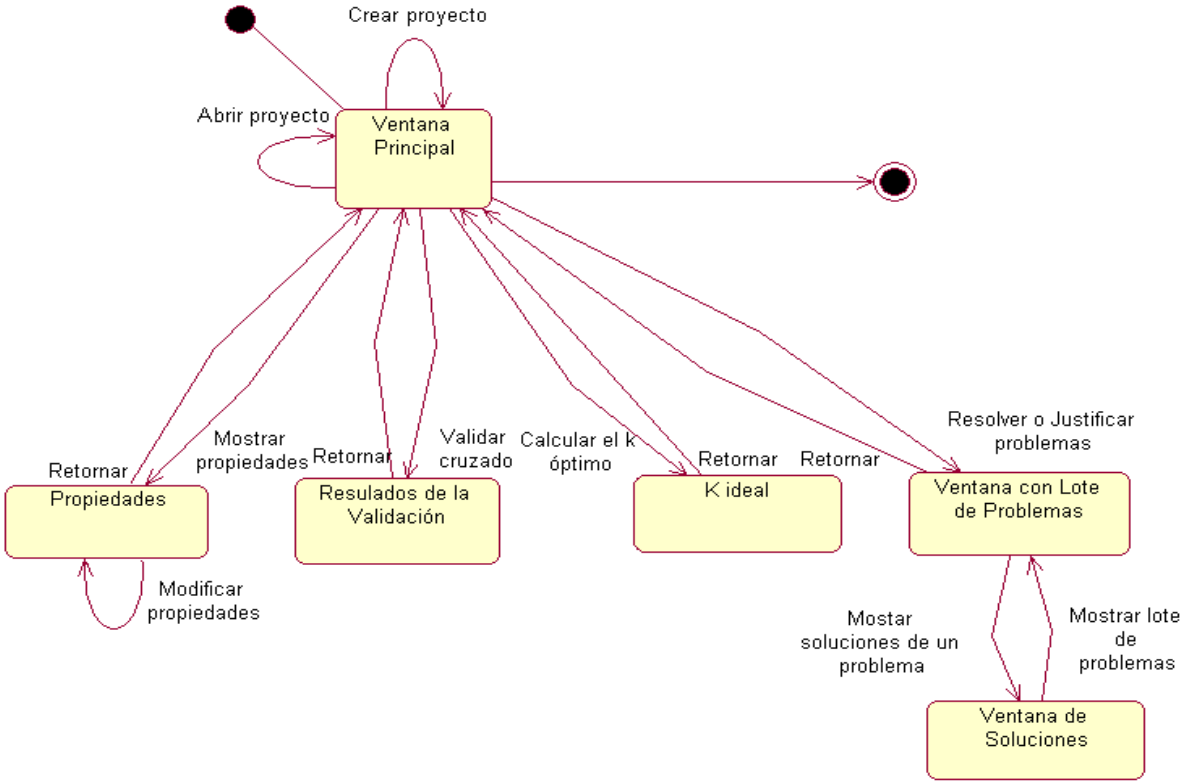


Figura 9. Diagrama de estado

Capítulo 3: Validación del módulo basado en casos de un modelo híbrido

Por otro lado, desde la perspectiva ingenieril y técnica, los SBC están listos para su uso en problemas de seguridad crítica como diagnósticos médicos. Debido a que estos pueden tomar decisiones críticas que los humanos en algunas situaciones no pueden, y además integrar información al proceso de toma de decisiones, es concebible que su uso pueda mejorar la salud y hasta salvar vidas. Sin embargo es prudente saber cuán seguro y efectivo es un programa antes de confiarle decisiones autónomas [BUC95].

Estas razones convierten a la evaluación en una etapa inviolable y de vital importancia en el desarrollo de cualquier software, y principalmente de los SBC. Por ejemplo, qué se debe evaluar en este tipo de sistemas, y cómo hacerlo se trata en [ROD03], donde el autor propone una metodología a seguir para hacer la evaluación de los sistemas desarrollados utilizando un modelo que combina el RBC con las RNA.

En el presente trabajo las validaciones a realizar están encaminadas a demostrar las siguientes hipótesis:

H_1: La extensión de la FS al tratamiento de rasgos continuos, utilizando conjuntos borrosos (*fuzzy sets approach*), muestra resultados superiores; con respecto a la FS del modelo original, donde este tipo de rasgo se trata como intervalo (*crisp set approach*).

H_2: Es factible utilizar un módulo basado en casos, que emplee una FS similar a la anterior, para proponer una solución a un problema.

3.1 ¿Cómo medir el desempeño de un modelo?

En este capítulo es importante definir cuándo decimos que una solución es correcta, considerando que un problema de los aquí tratados puede tener una solución compleja. Esto significa que se pueden tener varios rasgos objetivos a la vez, o alguno de ellos tomar más de un valor en un caso de la BC. Consideramos necesario explicar primeramente cómo se procede en la herramienta para evaluar una solución de este tipo.

Una muestra de *N* ejemplos caracterizados por *M* atributos, se puede representar como una matriz *NxM*, que denominaremos MA. El preprocesamiento de ésta (que denominaremos MA'), a partir

de la especificación de una función *f* para cada atributo siguiendo alguna de las variantes previamente definidas, da como resultado una matriz de dimensión *NxM'* donde:

$$M' = \sum_{i=1}^{/M/} m_i \qquad (17)$$

$$M = C_p \cup C_o$$

$$m_i = |Tx_i|, \forall x_i \in M$$

Donde*:*

C_p: conjunto de rasgos predictores.

C_o: conjunto de rasgos objetivos.

La solución a un problema (conjunto de valores dados a los rasgos objetivos) se puede ver como un vector m-dimensional, donde *m* se calcula de forma similar a *M'* en la expresión anterior pero considerando $M=C_o$.

Una medida de la similitud entre dos vectores soluciones *x, y* se puede estimar a partir de la siguiente expresión, que expresa el dual de una medida de distancia. Las variantes utilizadas fueron: la distancia de *Manhattan* o *City-Block* y la Euclideana [WIL97] (expresiones 19 y 20).

$$\delta(x, y) = 1 - \frac{Dist(x, y)}{m} \qquad (18)$$

$$Dist(x, y) = \sum_{i=1}^{m} |x_i - y_i| \qquad (19)$$

$$Dist(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \qquad (20)$$

3.2 Diseño de los experimentos

Con la finalidad de probar estas hipótesis, en los experimentos se utilizan 19 archivos de datos de la UCIMLR, cuyas características se muestran en el anexo 2. Nótese que es una muestra representativa de bases de casos a considerar, pues estos difieren en características tales como: ausencia de información en los datos, cantidad de ejemplos, cantidad de rasgos y naturaleza de los mismos, puede existir o no definición de las clases y en este último caso varían en cantidad, por mencionar algunas.

Utilizando cada archivo de datos se hizo una validación cruzada 10 particiones (*10 fold crossvalidation*), y en cada una el 70% se considera base de casos, y el resto muestra de control. Como resultado se mide para cada archivo el desempeño medio ($\breve{D}$) y la varianza del desempeño (σ^2), como se muestra en las expresiones 21 y 22, en las tres variantes que se explican a continuación:

- La componente basada en casos del modelo original [GAR96] *Modelo Duro*. En esta variante los atributos lineales se representan mediante intervalos utilizando el algoritmo CAIM [KUR04].
- La componente basada en casos en el nuevo modelo, extendiendo la función de semejanza del modelo original para utilizar conjuntos borrosos para modelar los atributos lineales. Los términos lingüísticos y las funciones de pertenencia que se definen para cada atributo lineal, se toman de los resultados expuestos en [VAR05]. Para este modelo se definen dos variantes: siguiendo el criterio de máxima membresía, que referenciamos como *Modelo Fuzzy (Max)*, y *Modelo Fuzzy (Std)* que significa considerar todos los términos lingüísticos.

$$\breve{D} = \frac{\sum_{i=1}^{n} D_i}{n} \qquad (21)$$

$$\sigma^2 = \frac{\sum_{i}^{n} \left(D_i - \breve{D}\right)^2}{n-1} \qquad (22)$$

Donde:

D_i: desempeño del modelo con la i-ésima partición.

Para demostrar la H_1, relacionada con la componente basada en casos para justificar, se debe demostrar que los k casos recuperados ante un problema P considerando la FS β (ver expresión 6), son en efecto los más similares de la BC a éste.

Una medida que cuantifique lo anterior se puede estimar considerando calculando para el problema P la coincidencia[2] entre las soluciones: obtenidas para este por la RNA ($P^r.sol$) y la que se sugiere en el contexto de los k casos seleccionados ($P^k.sol$) considerando la expresión 18.

[2] Utilizando la expresión 18

Considerando todos los problemas de la muestra de control, se obtendría entonces una medida del desempeño (*D*).

El procedimiento a seguir para demostrar la H_2 es similar al anterior, pero las soluciones a comparar son: la deseada para el problema, es decir, la que aparece para este en la muestra de control ($P^k.sol$), y la que se obtiene a partir de la componente basada en casos utilizando la FS β'.

3.3 Análisis de los resultados

Para el análisis de los resultados se utilizaron técnicas estadísticas, en particular comparación de poblaciones y estimación por intervalos de confianza.

Un esquema de comparación de poblaciones se utiliza para comparar una variable aleatoria entre dos poblaciones, siempre que esta sea continua o al menos ordinal. La comparación se puede hacer entre grupos independientes (comparaciones verticales o de muestras independientes) o en un mismo grupo entre dos momentos diferentes (comparaciones horizontales o de muestras apareadas). Desde este punto de vista se trata de precisar la posible dependencia de esa variable aleatoria continua respecto a una variable aleatoria discreta, concretamente dicotómica: el grupo ó el momento.

En el problema que nos ocupa estamos en presencia de una comparación de poblaciones de muestras apareadas, donde se comparan las variables anteriormente explicadas ($\check{D}$ y σ^2), considerando tres momentos diferentes en correspondencia con las tres variantes anteriormente definidas. Desde el punto de vista estadístico se formula la siguiente hipótesis:

H_o: Existe homogeneidad de los rangos medios

Si la significación de la prueba realizada es menor 0.05 (valor para la significación que se toma en este caso), se rechaza H_o con un 5% de error, es decir se considera que los rangos medios de la variable a comparar muestra diferencias significativas. En caso contrario, no se tienen elementos para rechazar H_o, y se concluye que no hay diferencias significativas.

Para probar esta hipótesis nos auxiliaremos del paquete estadístico *SPSS* (versión 9.0). Las pruebas a realizar son no paramétricas considerando el tamaño de la muestra. En particular utilizaremos primeramente la prueba de *"Friedman"* para tener un criterio inicial del comportamiento de la variable, y acode a estos resultados si se requiere un análisis más detallado utilizando la prueba de rangos de "*Wilcoxon"*

Por otro lado, resulta conveniente utilizar la estimación de parámetros por intervalos de confianza para propiciar un análisis cualitativo de los resultados alcanzados. En particular definiremos el intervalo de confianza de la variable *Ď*, utilizando la expresión 23.

Para probar la H_1 se midió el desempeño promedio (*Ď*) en las tres variantes definidas: *Modelo Duro*, *Modelo Fuzzy (Max)*, y *Modelo Fuzzy (Std).* Este valor y la varianza del desempeño, para los resultados de la validación realizada con cada uno de los 17 archivos de datos utilizados, se muestran en una tabla del anexo 3. Los resultados de la comparación de poblaciones, considerando estas variables, aparecen en el anexo 4. Su interpretación es la siguiente:

- La media del desempeño, en las tres variantes consideradas, no se diferencia de forma significativa (significación de la prueba: 0.373).
- Las varianza del desempeño, en las tres variantes consideradas, muestra diferencias significativas (significación de la prueba: 0.007). El orden entre el valor de la varianza es el siguiente: *Modelo Fuzzy (Max)*^{a} < *Modelo Duro*^{b} < *Modelo Fuzzy (Std)*^{b}, mostrando diferencias medianamente significativas entre el *Modelo Fuzzy (Max)*y los restantes.

Esto nos permite concluir sobre la H_1 que la extensión de la FS al tratamiento de rasgos continuos, utilizando conjuntos borrosos (*fuzzy set approach*) considerando el principio de máxima membresía, muestra resultados más estables y seguros en la BC consideradas; con respecto a la FS del modelo original, donde este tipo de rasgo se trata como intervalo (*crisp set approach*).

Para probar la H_2 se hacen dos tipos de experimentos. El primero de tipo cuantitativo, similar al descrito anteriormente para probar H_1, a partir de los datos que se muestran en la tabla del anexo 5 para los mismos archivos de datos. Los resultados del *SPSS* se muestran en el anexo 6, y su interpretación arroja que:

- La media del desempeño, en las tres variantes consideradas, no se diferencian de forma significativa (significación de la prueba: 0.166).
- La varianza del desempeño, en las tres variantes consideradas, no muestra diferencias significativas (significación de la prueba: 0.859).

Esto nos permite concluir sobre H_2, que es factible utilizar un módulo basado en casos con una FS, similar a la utilizada en el módulo basado en casos del modelo original para explicar la

solución, pero en este caso como resolvedor de problemas. Es decir, con esta nueva funcionalidad, las tres variantes muestran resultados similares.

El segundo experimento es de tipo cualitativo, y tiene como objetivo comparar el resultado de utilizar el módulo basado en casos como resolvedor de problemas en las diferentes variantes, con los resultados reportados en la literatura referenciada. Particularmente utilizaremos los resultados que se presentan en [WIL97], donde se realizó una validación similar a la realizada anteriormente con archivos de datos de la UCIMLR, con varias medidas tales como: HVDM, IVDM y WVDM que son modificaciones a la medida VDM.

Para ello utilizamos la estimación por intervalos de confianza. La idea es comparar el desempeño para cada archivo reportado en la literatura consultada y la media del desempeño obtenido con los experimentos realizados en cada variante, y que se muestran en el anexo 5. Si el valor del desempeño reportado en [WIL97] pertenece al intervalo de confianza calculado para $\check{D}$ en un archivo de datos, se considera que para este archivo el resultado es *comparable* (se representa por 0); si está a la derecha del intervalo, el resultado de la variante utilizada con este archivo de datos es *inferior* (se representa por -); y si queda por la izquierda es *superior* (se representa por +). A continuación se muestra una tabla con estos resultados para cada una de las variantes consideradas. Las tablas completas, que contienen la información sobre el intervalo de confianza, se muestran en el anexo 7.

RBC para resolver									
Base de casos	Modelo Duro			Modelo Fuzzy (Max)			Modelo Fuzzy (Std)		
	HVDM	IVDM	WVDM	HVDM	IVDM	WVDM	HVDM	IVDM	WVDM
Anneal	+	+	+	+	+	+	+	+	+
Cleveland	+	0	0	+	0	0	+	0	0
Credit-app	+	+	+	+	+	+	+	+	+
Diabetes	-	-	-	-	-	-	-	-	-
*Flag**	+	+	+	+	+	+	+	+	+
Glass	+	+	+	+	+	+	+	+	+
Hepatitis	+	0	0	+	0	0	+	0	+
Hypothyroid	+	-	0	+	-	+	+	0	-
Ionosphere	-	-	-	+	-	-	+	-	-

Iris	-	-	-	-	-	-	-	-	-
Liver Disorders	-	-	0	-	0	+	-	+	+
Sonar	-	-	-	-	-	-	-	-	-
Vehicle	+	+	+	+	+	+	+	+	+
Vowel	-	-	-	-	-	-	-	-	-
WBC	-	0	0	0	0	0	0	0	0
Wine	-	-	-	-	-	-	-	-	-
Zoo	-	-	-	-	-	-	-	-	-
Suma (+ y 0)	8	8	10	10	10	9	9	10	10

Nótese que las tres variantes muestran resultados comparables o superiores, con respecto a las medidas referenciadas, en al menos el 50% de los archivos de datos utilizados. El módulo basado en casos, que utiliza los conjuntos borrosos para modelar los atributos lineales, muestra discretamente mejores resultados.

Una característica común de los archivos de datos, con los que siempre el resultado es superior o comparable incluso en todas las variantes, es la ausencia de información. Esto se puede fundamentar en el hecho de que a diferencia de las medidas con las que se compara, la FS, tanto del modelo original como del nuevo modelo, toman la información de los pesos de una RNA.

Conclusiones

En el presente trabajo se implementó la componente basada en casos del modelo híbrido expuesto en [GAR00], como una extensión del modelo original que se presentó en [GAR96]. La misma incorpora el tratamiento de rasgos continuos utilizando conjuntos borrosos, y permite utilizar la componente basada en casos con dos funcionalidades: para justificar la solución propuesta por la RNA y como resolvedor de problemas. La herramienta desarrollada brinda facilidades para la validación del módulo basado en casos implementado, el cual fue validado utilizando 19 archivos de datos de la UCIMLR. Además el módulo basado en casos implementado queda disponible en forma de biblioteca de funciones para su fácil incorporación a la versión final de la herramienta que implementa el nuevo modelo.

A los efectos de justificar una solución se permite entrar un muestra resuelta por la RNA, y mostrar los *k* casos más similares a cada uno de los problemas que la conforman. Además, se determina cuál de estos casos es el más relevante, el cual se considera a los efectos de cuantificar una medida de desempeño del módulo basado en casos en este rol de justificador. La validación de los resultados muestra que con la extensión de la FS al tratamiento de rasgos continuos, utilizando conjuntos borrosos (*fuzzy set approach*) y considerando el principio de máxima membresía, se obtienen resultados más estables y seguros en las BC consideradas; que utilizando intervalos (*crisp set approach*).

Para resolver un problema se implementa una variante para calcular el *LOOCE*, a partir del cual se calcula para una BC el número *k* óptimo de casos más similares a considerar para resolver un problema. Para recuperar estos casos se utiliza una FS similar a la implementada en la variante para justificar. A los efectos de proponer una solución se explora un conjunto de soluciones formado a partir de las soluciones presentes en los casos más similares recuperados. La validación de estos resultados muestra que es factible utilizar un módulo basado en casos con una FS, similar a la utilizada en el módulo basado en casos del modelo original para explicar la solución, pero en este caso como resolvedor de problemas. Siguiendo los dos enfoques: duro y *fuzzy* en la FS, se obtienen resultados comparables o superiores en al menos el 50% de los archivos de datos utilizados, respecto a resultados reportados en la literatura para otras medidas de similitud. La variante *fuzzy* muestra discretamente mejores resultados.

Recomendaciones

Como recomendaciones de este trabajo se proponen las siguientes:

1- Comparar los resultados obtenidos con la componente basada en casos en el rol de resolvedor, con los resultados de la RNA.

2- Ampliar la validación realizada utilizando otras bases de casos, que se caractericen por: tener más de un rasgo objetivo, rasgos objetivos que puedan tener presente en un caso más de un valor a la vez y donde haya ausencia de información en los casos.

3- Estudiar las características de las bases de casos con las cuales la componente basada en casos del modelo híbrido muestra mejor desempeño que otros modelos similares.

Referencias Bibliográficas

[BEL02] Bello, P.R. et. Al. Aplicaciones de la Inteligencia Artificial. Ed. Universidad de Guadalajara. 2002.

[BUC95] Buchanan, B. G. Report from Workshop on Evaluation of Knowledge-based Systems, 1995.

[CAS00] Castellano G. et. al.: Fuzzy inference and rule extraction using a neural network. Neural Network World, Volume 10. 2000.

[CAU91] Caudill, M., Expert Networks, Byte. (Oct 1991).

[CAU94] M. Caudill, Expert Networks, Byte. Oct 1991.

[COR00] F. Fdez-Riverola; J.M. Corchado (2000) Sistemas híbridos Neuro-Simbólicos: Una revisión. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No 11, Vol I, pp. 12-26.

[CHI96] Chin-Teng Lin y C.S George Lee Neural Fuzzy Systems: a neuro-fuzzy synergism to intelligent systems. Prentice Hall. 1996.

[GAR00] García, M.; Rodríguez, Y.; Bello, P. Usando conjuntos borrosos para implementar un modelo para sistemas basados en casos interpretativos. In Proceedings of IBERAMIA-SBIA 2000, Eds por M. C. Monard y J.S. Sichman, Sao Paulo, Brasil, Nov. 2000.

[GAR03] García, M., et. al. Redes Neuronales Artificiales, ISBN 970-27-0409-X Prometeo Editores. México. 2003.

[GAR96] García, MM and Bello, P.R.. A model and its different applications to case-based reasoning. Knowledge-based systems 9 1996 465-473.

[GIA92] Giacometti, A., An hybrid approach to computer-aided diagnosis in electromyography, in: Proceeding of the 14th Annual International Conference of IEEE Engineering in Medicine and Biology Society (1992).

[GON93] González, A.V. The Engineering of Knowledge-based systems Theory and Practice. Prentice hall, New Jersey .1993.

[GUA94] Guardati, S.Razonamiento Basado en Casos. Soluciones avanzadas. Ano 2 Nro 13. Sept 1994.

[HAT00] Hatzilygeroudis, I. et. al.: Neurules: Improving the performance of symbolic rules. International Journal on Artificial Intelligence Tools (IJAIT), vol. 9,

No. 1. 2000.

[HED93] Hedberg, S., New knowledge tools, Byte (July 1993).

[HIL95] Hilera, José R. y Martínez, Victor J.,Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones. Madrid, España.1995.

[JUR93] Jurisica, I. Representation and Management Issues for Case-Based Reasoning Systems Department of Computer Science, University of Toronto, Toronto, Ontario M5S1A4, Canadá, Sept 1993.

[JYH98] Jyh-Shing, R. et. al. Neuro-Fuzzy and Soft Computing. Prentice Hall 1998.

[KUR04] Kurgan L.; et. al.: CAIM Discretization Algorithm. IEEE Transactions on Knowledge and Data Engineering. Vol 16. No. 2 (2004)

[LEE02] B. Lees and J. Corchado. Integrated Case-based Neural Network Approach to Problem Solving. Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence (LNAI 1570) Springer Verlag ISSN 0302-9743. ISBN: 3-540-65658-8. pag. 157-166. 2002.

[LIE93] Liebowitz J., Roll your own hybrids, Byte (July 1993).

[MAL01] Maria Malek, Rushed Kanawati: A Cooperating Hybrid Neural-CBR Classifiers for Building On-Line Communities, Workshop on Soft Computing in Case-Based Reasoning, International Conference on Case-Based Reasoning (ICCBR'01). 2001.

[MAT01] Matich, Damián Jorge. Redes neuronales. Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional. Marzo, 2001.

[MCC89] McClelland, J.L. y Rumelhart, D.E. Explorations in parallel distributed processing.MIT Press, 1989.

[MED92] Medsker L. R. y Bailey D. L.(1992). Models and guidelines for integrating expert systems and neural networks. Hybrid Architectures for Intelligent Systems. Ed. Kandel A. y Langholz G. CRC Press, Boca Raton.

[ROD02] Rodríguez, Y. Sistemas Basados en el Conocimiento. En Aplicaciones de la Inteligencia Artificial, R.Bello y otros, Editorial La Noche. ISBN 970-27-0177-5. México, 2002.

[ROD03] Rodríguez, Y., et. al. Sistema Computacional para la determinación de propiedades anticancerigenas en el diseño de un fármaco. Memorias de Informática 2003 ISBN: 959237095-8

[RUI93] Ruiz, J. and Lazo, M. Modelos Matemáticos para el Reconocimiento de Patrones.
Edit UCLV. Santa Clara. Cuba. 1993.

[SAN01] Sankar K.Pal, et. al. Soft Computing in Case Based Reasoning. Edit. Springer-Verlag London. 2001.

[SHI93] Shishkov y R.I. Trifonov, A hybrid neural production system, in: Proceeding of World on Neural Networks. 1993.

[SIM95] Sima, J., Neural Expert Systems, Neural Networks. Vol. 8 #2. 1995.

[SOV01] Ricardo B. Sovat, Andre C.P.L.F. de Carvalho, 2001: Retrieval and Adaptation of Cases Using an Artificial Neural Network, Workshop on Soft Computing in Case-Based Reasoning, International Conference on Case-Based Reasoning (ICCBR'01).

[STA86] [Stanfill & Waltz, 86] Stanfill, C. and Waltz, D.. Toward memory-based reasoning. Comm. of ACM, 29 (1986) 1213-1228.

[TOW94] Towell, G.G. y Sharlik, J.W., Knowledge-based artificial neural networks, Artificial Intelligence. 70 (1994) 119-165.

[TOW94] G.G. Towell y J.W. Sharlik, Knowledge-based artificial neural networks, Artificial Intelligence 70 1994 119-165.

[VAR05] Varela, Alain J., et.al. Construcción automatizada de Funciones de Pertenencia. Trabajo de Diploma. UCLV, 2005.

[WET97] Wettschereck D., Ahad D. W. y Morí T.. A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms. Artificial Intelligence Review 11, 273-314, 1997.

[WIL00] Wilson, D. Randall y Martínez Tony R.. Computacional Intelligence, volumen 16, Number 1, pp. 1-28, 200

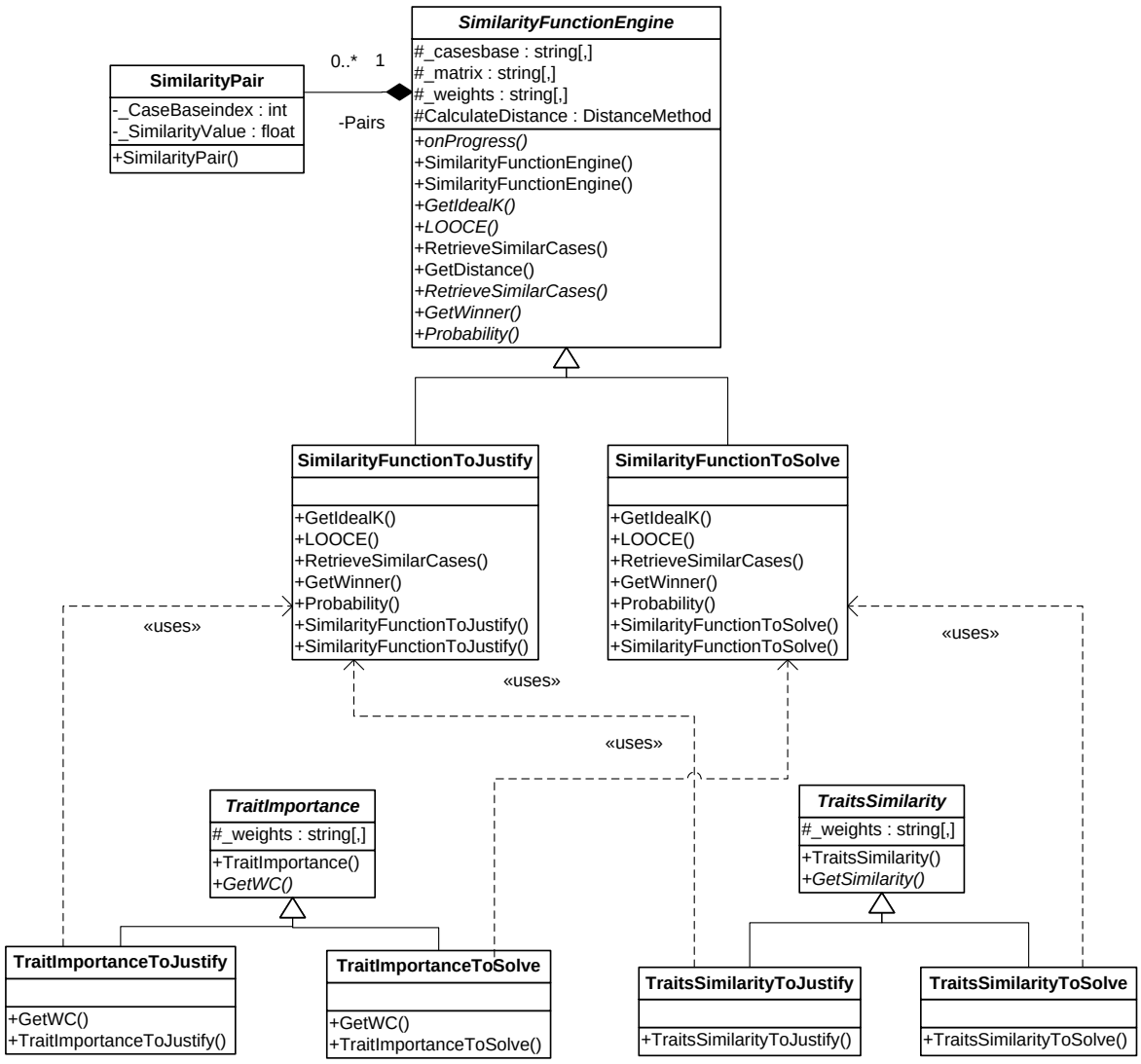
[WIL97] Wilson, D. Randall y Martínez Tony R.. Journal of Artificial Intelligence Research 6 (1997) 1-34.

[ZAD76] Zadeh L.. The concept of a linguistic variable and its application to approximate reasoning. Information Science 9, page. 43-80,1976.

[ZUR92] Zurada, J. M., Introduction to Artificial Neural Systems, West Publishing, 1992.

Anexos

Anexo 1: Diagrama de clases



Anexo 2: Descripción de los archivos de datos utilizados en los experimentos

Archivo de datos	Cantidad de ejemplos	Características de los rasgos predictores		Tiene ausencia de información	Cantidad de clases
		Simbólicos	Lineales		
Anneal	898	25	5	Si	6
Cleveland	303	9	5	Si	2
Credit-app	690	9	6	Si	2
Diabetes	768	0	8	No	2
Flag(multi)	194	18	6	No	*
Glass	214	0	9	No	7
Hepatitis	155	13	6	Si	2
Hypothyroid	3772	16	6	Si	4
Ionosphere	351	0	34	No	2
Iris	150	0	4	No	3
Liver-disorders	345	0	6	No	2
Segmentation	2310	0	19	No	7
Solar-flare	1389	3	10	No	*
Sonar	208	0	60	No	2
Vehicle	846	0	18	No	4
Vowel	990	1	10	No	11
WBC	699	0	9	No	2
Wine	178	0	13	No	3
Zoo	101	16	1	No	7

(* Significa que no está predefinida una partición de los rasgos en predictores y objetivos)

Anexo 3: Resultados de la validación cruzada (RBC para justificar)

Base de casos	Modelo Duro		Modelo Fuzzy (Max)		Modelo Fuzzy (Std)	
	$\check{D}$	Σ^2	$\check{D}$	σ^2	$\check{D}$	σ^2
Anneal	0.978815	0.000021	0.978815	0.000021	0.979704	0.000068
Cleveland	0.818083	0.008404	0.821646	0.007851	0.821646	0.007851
Credit-app	0.814976	0.000493	0.813527	0.000240	0.797101	0.000902
Diabetes	0.657143	0.000507	0.650649	0.000371	0.628139	0.001226
Flag	0.808248	0.000258	0.818192	0.000075	0.820452	0.000435
Glass	0.873334	0.000322	0.864876	0.000096	0.866763	0.000436
Hepatitis	0.825532	0.001187	0.812766	0.002595	0.82766	0.001957
Hypothyroid	0.971059	0.000005	0.970317	0.000002	0.969116	0.000008
Ionosphere	0.740566	0.008826	0.887736	0.000207	0.9	0.001032
Iris	0.808889	0.00168	0.841481	0.001514	0.832593	0.003270
Liver Disorders	0.569231	0.001311	0.588462	0.001557	0.600962	0.001895
Segment	0.937003	0.000010	0.910824	0.000030	0.905382	0.000032
Solar Flare	0.964229	0.000014	0.985551	0.000003	0.985351	0.000001
Sonar	0.526984	0.003124	0.636508	0.001705	0.638095	0.000493
Vehicle	0.825	0.000303	0.816929	0.000052	0.824606	0.000154
Vowel	0.908571	0.000122	0.797246	0.000085	0.842211	0.000106
WBC	0.95867	0.001126	0.95716	0.000532	0.95716	0.000532
Wine	0.920988	0.000549	0.84321	0.000747	0.848148	0.000781
Zoo	0.980451	0.000000	0.980451	0.000000	0.980451	0.000000

Anexo 4: Resultados del SPSS con el módulo basado en casos para justificar

NPar Tests

Friedman Test

Ranks

	Mean Rank
Desempeño medio del Modelo Duro	2.09
Desempeño medio en el Modelo Fuzzy (Max)	1.74
Desempeño medio en el Modelo Fuzzy (Std)	2.18

Test Statistics[a]

N			17
Chi-Square			2.066
df			2
Monte Carlo Sig.	Sig.		.373
	99% Confidence Interval	Lower Bound	.360
		Upper Bound	.385

a. Friedman Test

NPar Tests

Friedman Test

Ranks

	Mean Rank
Varianza del desempeño en el Modelo Duro	2.15
Varianza del desempeño en el Modelo Fuzzy (Max)	1.44
Varianza en el desempeño del Modelo Fuzzy (Std)	2.41

Test Statistics[a]

N			17
Chi-Square			9.541
df			2
Monte Carlo Sig.	Sig.		.007
	99% Confidence Interval	Lower Bound	.005
		Upper Bound	.009

a. Friedman Test

NPar Tests

Wilcoxon Signed Ranks Test

Ranks

		N	Mean Rank	Sum of Ranks
Varianza del desempeño en el Modelo Duro - Varianza del desempeño en el Modelo Fuzzy (Max)	Negative Ranks	3[a]	9.00	27.00
	Positive Ranks	12[b]	7.75	93.00
	Ties	2[c]		
	Total	17		

a. Varianza del desempeño en el Modelo Duro < Varianza del desempeño en el Modelo Fuzzy (Max)

b. Varianza del desempeño en el Modelo Duro > Varianza del desempeño en el Modelo Fuzzy (Max)

c. Varianza del desempeño en el Modelo Fuzzy (Max) = Varianza del desempeño en el Modelo Duro

Test Statistics[b,c]

			Varianza del desempeño en el Modelo Duro - Varianza del desempeño en el Modelo Fuzzy (Max)
Z			-1.874[a]
Asymp. Sig. (2-tailed)			.061
Monte Carlo Sig. (2-tailed)	Sig.		.063
	99% Confidence Interval	Lower Bound	.054
		Upper Bound	.072

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

c. Based on 10000 sampled tables with starting seed 926214481.

Wilcoxon Signed Ranks Test 2

Ranks

		N	Mean Rank	Sum of Ranks
Varianza en el desempeño del Modelo Fuzzy (Std) - Varianza del desempeño en el Modelo Duro	Negative Ranks	6[a]	9.67	58.00
	Positive Ranks	10[b]	7.80	78.00
	Ties	1[c]		
	Total	17		

a. Varianza en el desempeño del Modelo Fuzzy (Std) < Varianza del desempeño en el Modelo Duro

b. Varianza en el desempeño del Modelo Fuzzy (Std) > Varianza del desempeño en el Modelo Duro

c. Varianza del desempeño en el Modelo Duro = Varianza en el desempeño del Modelo Fuzzy (Std)

Test Statistics[b,c]

			Varianza en el desempeño del Modelo Fuzzy (Std) - Varianza del desempeño en el Modelo Duro
Z			-.517[a]
Asymp. Sig. (2-tailed)			.605
Monte Carlo Sig. (2-tailed)	Sig.		.631
	99% Confidence Interval	Lower Bound	.607
		Upper Bound	.655

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

c. Based on 10000 sampled tables with starting seed 926214481.

Wilcoxon Signed Ranks Test 3

Ranks

		N	Mean Rank	Sum of Ranks
Varianza en el desempeño del Modelo Fuzzy (Std) - Varianza del desempeño en el Modelo Fuzzy (Max)	Negative Ranks	2[a]	11.00	22.00
	Positive Ranks	12[b]	6.92	83.00
	Ties	3[c]		
	Total	17		

a. Varianza en el desempeño del Modelo Fuzzy (Std) < Varianza del desempeño en el Modelo Fuzzy (Max)

b. Varianza en el desempeño del Modelo Fuzzy (Std) > Varianza del desempeño en el Modelo Fuzzy (Max)

c. Varianza del desempeño en el Modelo Fuzzy (Max) = Varianza en el desempeño del Modelo Fuzzy (Std)

Test Statistics[b,c]

			Varianza en el desempeño del Modelo Fuzzy (Std) - Varianza del desempeño en el Modelo Fuzzy (Max)
Z			-1.915[a]
Asymp. Sig. (2-tailed)			.056
Monte Carlo Sig. (2-tailed)	Sig.		.058
	99% Confidence Interval	Lower Bound	.049
		Upper Bound	.066

a. Based on negative ranks.

b. Wilcoxon Signed Ranks Test

c. Based on 10000 sampled tables with starting seed 926214481.

Anexo 5: Resultados de la validación cruzada (RBC para resolver)

Base de casos	Modelo Duro		Modelo Fuzzy (Max)		Modelo Fuzzy (Std)	
	$\check{D}$	σ^2	$\check{D}$	σ^2	$\check{D}$	σ^2
Anneal	0.985037	0.000011	0.983111	0.000013	0.984148	0.000043
Cleveland	0.818205	0.010572	0.81498	0.010298	0.81498	0.010298
Credit-app	0.850242	0.000679	0.841546	0.000310	0.830435	0.000837
Diabetes	0.659307	0.000554	0.648485	0.000536	0.628139	0.001110
Flag	0.779887	0.000169	0.774463	0.000095	0.772655	0.000092
Glass	0.867693	0.000297	0.868868	0.000319	0.865884	0.000234
Hepatitis	0.814894	0.001313	0.810638	0.003164	0.83617	0.002721
Hypothyroid	0.969717	0.000004	0.969999	0.000003	0.968939	0.000007
Ionosphere	0.740566	0.008826	0.886792	0.000218	0.89717	0.001176
Iris	0.885926	0.003563	0.903704	0.000841	0.871111	0.002636
Liver Disorders	0.568269	0.001283	0.588462	0.001557	0.604808	0.001797
Segment	0.940549	0.000011	0.924182	0.000019	0.916719	0.000040
Solar Flare	0.964109	0.000014	0.98281	0.000003	0.982663	0.000001
Sonar	0.526984	0.003124	0.622222	0.002284	0.631746	0.000885
Vehicle	0.834252	0.000111	0.83248	0.000514	0.832284	0.000154
Vowel	0.87383	0.000173	0.791737	0.000051	0.822621	0.000156
WBC	0.960099	0.001006	0.955814	0.001187	0.955814	0.001187
Wine	0.91358	0.00044	0.838272	0.000625	0.862963	0.000422
Zoo	0.980322	0.000000	0.980322	0.000000	0.980322	0.000000

Anexo 6: Resultados del SPSS con el módulo basado en casos para resolver

NPar Tests

Friedman Test

Ranks

	Mean Rank
Desempeño medio del Modelo Duro	2.35
Desempeño medio en el Modelo Fuzzy (Max)	1.88
Desempeño medio en el Modelo Fuzzy (Std)	1.76

Test Statistics[a]

N			17
Chi-Square			3.613
df			2
Monte Carlo Sig.	Sig.		.166
	99% Confidence Interval	Lower Bound	.156
		Upper Bound	.176

a. Friedman Test

NPar Tests

Friedman Test

Ranks

	Mean Rank
Varianza del desempeño en el Modelo Duro	2.06
Varianza del desempeño en el Modelo Fuzzy (Max)	1.88
Varianza en el desempeño del Modelo Fuzzy (Std)	2.06

Test Statistics[a]

N			17
Chi-Square			.387
df			2
Monte Carlo Sig.	Sig.		.859
	99% Confidence Interval	Lower Bound	.850
		Upper Bound	.868

a. Friedman Test

Anexo 7: Intervalos de confianza para los desempeños en cada variante

RBC para resolver							
Base de casos	Modelo Duro		Intervalo de confianza		Observaciones		
	$\check{D}$	σ^2	$\alpha = 0.05$		HVDM	IVDM	WVDM
Anneal	0.985037	0.000011	0.984344	0.98573	+	+	+
Cleveland	0.818205	0.010572	0.779812	0.856599	+	0	0
Credit-app	0.850242	0.000679	0.84398	0.856503	+	+	+
Diabetes	0.659307	0.000554	0.653965	0.66465	-	-	-
*Flag**	0.779887	0.000169	0.773626	0.786148	+	+	+
Glass	0.867693	0.000297	0.859849	0.875536	+	+	+
Hepatitis	0.814894	0.001313	0.795587	0.834201	+	0	0
Hypothyroid	0.969717	0.000004	0.969527	0.969907	+	-	0
Ionosphere	0.740566	0.008826	0.708295	0.772838	-	-	-
Iris	0.885926	0.003563	0.852871	0.918981	-	-	-
Liver Disorders	0.568269	0.001283	0.555965	0.580574	-	-	0
Sonar	0.526984	0.003124	0.501541	0.552427	-	-	-
Vehicle	0.834252	0.000111	0.83198	0.836524	+	+	+
Vowel	0.87383	0.000173	0.871204	0.876456	-	-	-
WBC	0.960099	0.001006	0.952538	0.96766	-	0	0
Wine	0.91358	0.00044	0.903145	0.924015	-	-	-
Zoo	0.980322	0.000000	0.980079	0.980565	-	-	-
Total (+ y 0)					8	8	10

RBC para resolver							
Base de casos	Max		Intervalo de confianza		Observaciones		
	$\check{D}$	σ^2	$\alpha = 0.05$		HVDM	IVDM	WVDM
Anneal	0.983111	0.000013	0.982348	0.983874	+	+	+
Cleveland	0.81498	0.010298	0.777088	0.852872	+	**0**	**0**
Credit-app	0.841546	0.000310	0.837315	0.845776	+	+	+
Diabetes	0.648485	0.000536	0.643228	0.653742	-	-	-
*Flag**	0.774463	0.000095	0.769756	0.779171	+	+	+
Glass	0.868868	0.000319	0.860742	0.876995	+	+	+
Hepatitis	0.810638	0.003164	0.780666	0.840611	+	**0**	**0**
Hypothyroid	0.969999	0.000003	0.969835	0.970163	+	-	+
Ionosphere	0.886792	0.000218	0.881726	0.891859	+	-	-
Iris	0.903704	0.000841	0.887641	0.919767	-	-	-
Liver Disorders	0.588462	0.001557	0.574905	0.602018	-	0	+
Sonar	0.622222	0.002284	0.600466	0.643978	-	-	-
Vehicle	0.83248	0.000514	0.827592	0.837369	+	+	+
Vowel	0.791737	0.000051	0.790319	0.793154	-	-	-
WBC	0.955814	0.001187	0.947599	0.96403	0	0	0
Wine	0.838272	0.000625	0.82584	0.850703	-	-	-
Zoo	0.980322	0.000000	0.980079	0.980566	-	-	-
Total (+ y 0)					10	9	10

RBC para resolver							
Base de casos	Std		Intervalo de confianza		Observaciones		
	$\check{D}$	σ^2	$\alpha = 0.05$		HVDM	IVDM	WVDM
Anneal	0.984148	0.984148	0.982776	0.985521	+	+	+
Cleveland	0.81498	0.81498	0.777088	0.852872	+	0	0
Credit-app	0.830435	0.830435	0.823484	0.837386	+	+	+
Diabetes	0.628139	0.628139	0.620578	0.635699	-	-	-
*Flag**	0.772655	0.000092	0.768036	0.7772751	+	+	+
Glass	0.865884	0.865884	0.858926	0.872841	+	+	+
Hepatitis	0.83617	0.83617	0.808373	0.863967	+	0	+
Hypothyroid	0.968939	0.968939	0.968667	0.969211	+	0	-
Ionosphere	0.89717	0.89717	0.885391	0.908949	+	-	-
Iris	0.871111	0.871111	0.842678	0.899544	-	-	-
Liver Disorders	0.604808	0.604808	0.590247	0.619368	-	+	+
Sonar	0.631746	0.631746	0.618207	0.645285	-	-	-
Vehicle	0.832284	0.832284	0.829607	0.83496	+	+	+
Vowel	0.822621	0.822621	0.820127	0.825115	-	-	-
WBC	0.955814	0.955814	0.947599	0.96403	0	0	0
Wine	0.862963	0.862963	0.852751	0.873175	-	-	-
Zoo	0.980322	0.980322	0.980079	0.980566	-	-	-
Total					10	10	9