UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TITULO: APLICACIÓN DE LA TEORÍA DE LOS CONJUNTOS AROXIMADOS EN EL PREPROCESAMIENTO DE LOS CONJUNTOS DE ENTRENAMIENTO PARA ALGORITMOS DE APRENDIZAJE AUTOMATIZADO

Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas

Autor: MSc. Yailé Caballero Mota
Tutor: Dr. Rafael Bello Pérez

Santa Clara
2007

SÍNTESIS

En el Aprendizaje Automatizado es una necesidad el preprocesamiento de la información. La Teoría de los Conjuntos Aproximados (RST) abrió una nueva dirección en el desarrollo de teorías sobre la información incompleta y es una poderosa herramienta para el análisis de datos. En esta investigación se demuestra la posibilidad de usar esta teoría en el preprocesamiento de los datos, tanto para encontrar conjuntos reducidos de atributos y editar conjuntos de entrenamiento para resolver problemas de clasificación supervisada, como para generar conocimiento a priori sobre un conjunto de datos. Se proponen dos algoritmos para obtener reductos. Uno de ellos integra funciones de adaptabilidad con algoritmos basados en estimación de distribuciones. El otro, construye un reducto a través de funciones heurísticas que combinan criterios de relevancia de los atributos. Para la edición de conjuntos de entrenamiento, se proponen dos nuevos algoritmos basados en los conceptos de aproximación de la RST. Además, se desarrolla una propuesta para caracterizar a priori conjuntos de entrenamiento, usando medidas de estimación de la RST.

Los métodos propuestos han sido estudiados experimentalmente usando bases de datos internacionales; así como su aplicación en el preprocesamiento de los datos para pronosticar, de forma automatizada, las temperaturas diarias en el Centro Meteorológico de Camagüey.

ÍNDICE

INTRODUCCIÓN

“Ahora que hemos reunido tantos datos ¿qué hacemos con ellos?”

Esta pregunta es común en numerosas áreas, debido a que la revolución digital ha facilitado la captura de la información y su almacenamiento no es costoso. Pero, ¿para qué se almacenan los datos? Para los científicos los datos representan observaciones cuidadosamente recogidas de algún fenómeno en estudio; en los negocios, los datos guardan informaciones sobre mercados, competidores y clientes; en procesos industriales, recogen valores sobre el cumplimiento de objetivos. El verdadero valor de los datos radica en la posibilidad de extraer de ellos información útil para la toma de decisiones o la exploración y comprensión de los fenómenos que le dieron lugar (Ruiz 2006). El análisis de datos es importante en ramas como: bioinformática, medicina, economía y finanzas, industria, medio ambiente, entre otras, donde el preprocesamiento de estos es esencial. Precisamente, el pronóstico del tiempo constituye un problema real donde el análisis de datos cobra una alta significación, dada por la complejidad y variabilidad de este proceso a lo largo de los años.

En la actualidad, los pronósticos del tiempo se han convertido en un fenómeno corriente y el servicio que brindan los diferentes Centros Meteorológicos ha adquirido enorme importancia para diferentes ramas de la economía del país. La automatización de procesos para el pronóstico meteorológico cobra un interés extraordinario en los últimos tiempos por investigadores de diversas ramas. Dentro del pronóstico del tiempo, la predicción automatizada de los valores de las temperaturas es un problema real a resolver, que merece una atención especial. El Centro Meteorológico de Camagüey, como muchos otros, ha tratado de pronosticar automáticamente los valores diarios de las temperaturas máximas y mínimas, como elemento imprescindible para el pronóstico del tiempo por parte de los especialistas de esta área. Se acumula un gran volumen de datos en cuanto a temperaturas y variables asociadas a estas, de los años 2000 hasta el 2006; sin embargo, con los métodos actuales de automatización del pronóstico no se han obtenido los resultados esperados, hasta ahora.

Antecedentes y actualidad del tema

El proceso completo de extraer conocimiento a partir de bases de datos se conoce como KDD (Knowledge Discovery in Databases). Este proceso comprende diversas etapas, que van desde la obtención de los datos hasta la aplicación del conocimiento adquirido en la toma de decisiones. Entre esas etapas, se encuentra la que puede considerarse como el núcleo del proceso KDD y que consiste en la extracción del conocimiento a partir de los datos. Esta fase es crucial para la obtención de resultados apropiados, y depende del algoritmo de aprendizaje automatizado que se aplique, no obstante, se ve influida en gran medida por la calidad de los datos que llegan desde la fase previa (Ruiz 2006).

En los sistemas de aprendizaje, la inducción se usa frecuentemente. Entre los métodos de aprendizaje a partir de ejemplos están las Redes Neuronales Artificiales (RNA) (Rosemblatt 1962), el método de los k-Vecinos más Cercanos (k-NN) (Cover and Hart 1967), el algoritmo C4.5 (Quinlan 1993), entre otros. La clasificación supervisada puede resultar en la práctica, tediosa y en ocasiones inaplicable si se dispone de conjuntos de entrenamiento numerosos y con datos de alta dimensionalidad. Estos dos factores determinan el costo computacional de la aplicación de la mayoría de los métodos de clasificación supervisada existentes.

Existen diferentes estrategias que permiten la aplicación de los métodos de clasificación supervisada con un costo computacional menor. Estas estrategias pueden agruparse en dos, de acuerdo a la filosofía en que se basan: i) reducir el conjunto de entrenamiento sin sacrificar demasiado la calidad de la clasificación supervisada y ii) mejorar la eficiencia computacional del método de aprendizaje.

Por ello, desde la década de los sesenta, las investigaciones relacionadas con la selección de atributos intentan reducir el espacio de hipótesis de conjuntos de datos en tareas concretas, en un intento de encontrar subconjuntos de atributos que satisfagan una o ambas estrategias (Dimitriev, Zhuravlev et al. 1966; Wróblewski 1995; Dash and Liu 2003; Ding and Peng 2003; Guyon and Elissee 2003; Jensen and Qiang 2003; Santiesteban and Pons 2003; Liu and Yu 2005). Por su parte, los métodos de edición de prototipos se basan en la selección de un subconjunto del conjunto de entrenamiento que satisfaga también esas estrategias

(Hart 1968; Devijver and Kittler 1980; Aha, Kibler et al. 1991; Brighton and Mellish 2002; Jiang and Zhou 2004; García and Shulcloper 2005; Olvera, Carrasco et al. 2005).

Una de las teorías más recientes empleada para el análisis de datos es la Teoría de los Conjuntos Aproximados RST (Rough Set Theory) (Pawlak 1982; Komorowski and Pawlak 1999; Greco 2001; Pal 2002; Segovia 2003; Tay and Shen 2003; Yao 2003; Caballero 2005; Mitra 2005; Bello, Nowe et al. 2006). Esta teoría se considera como una de las cinco áreas claves y no tradicionales de la Inteligencia Artificial y de la Teoría de la Información Incompleta, pues constituye una herramienta muy útil para el manejo de la información no completa o imprecisa (Chin, Liang et al. 2003; Peters 2005; Skowron, Swiniarski et al. 2005; Slezak 2005; Wolski 2005). La RST se ha usado para la generación de reglas (Koczkodaj 1998; Ohrn, Komorowski et al. 1998; Lee, Propes et al. 2002; Tsumoto 2003; Grzymala-Busse and Siddhaye 2004), la selección de atributos (Pawlak 1995a; Koczkodaj 1998; Yao and Wong 1999; Dunstsh and Gunter 2000; Zhong, Dong et al. 2001), entre otras aplicaciones (Hu, Lin et al. 2003; Pal and Mitra 2003; Zheng, Wang et al. 2003; Bazan and Szczuka 2005; Hor and Crossley 2005; Kostek, Szczuko et al. 2005; Suraj and Grochowalski 2005; Kierczak, Rudnicki et al. 2006; Revett, Gorunesco et al. 2006; Caballero, Bello et al. 2007).

Formulación del problema

El desarrollo de los sistemas inteligentes y la generación creciente de información en soporte electrónico, han conducido al auge de las técnicas de aprendizaje automatizado. Estas técnicas se basan en el empleo de la información sobre un dominio de aplicación como un conjunto de entrenamiento (CE), desde donde el método de aprendizaje extrae el conocimiento para resolver problemas de ese dominio. La calidad de ese conjunto es vital para la calidad del conocimiento descubierto, por lo que su análisis es un área de investigación relevante en el campo del aprendizaje automatizado. Entre las acciones principales comprendidas en el análisis del CE están la estimación de la calidad y el mejoramiento de los mismos, que incluye la selección de atributos y la edición de los CE. Todo ello constituye una problemática a la cual aún la ciencia no ha dado respuestas definitivas, por lo que se formula el siguiente **problema de investigación**:

Los métodos existentes actualmente no han resuelto totalmente la selección de atributos y la edición de CE con un costo computacional adecuado y con frecuencia estos tienen un comportamiento diferente en cada dominio de aplicación. Esto afecta a los algoritmos de aprendizaje automatizado encargados de extraer el conocimiento inherente a los datos.

Por otra parte, los métodos para estudiar la calidad de los CE usualmente son aplicados post-aprendizaje, de modo que tienen incluido el costo computacional de aplicar el método de aprendizaje; encontrar métodos que permitan una evaluación a priori del CE es un problema altamente relevante en al área del aprendizaje automatizado.

El **objetivo general** de esta investigación es:

Diseñar métodos para el mejoramiento de los conjuntos de entrenamiento, tanto para encontrar un conjunto reducido de atributos y editar conjuntos de entrenamiento para resolver problemas de clasificación supervisada, como para caracterizar a priori un conjunto de entrenamiento; a través del uso de la Teoría de los Conjuntos Aproximados.

Para lograr este objetivo se plantean los siguientes **objetivos específicos** de la investigación:

1. Diseñar nuevos métodos para la selección de atributos relevantes, que permitan disminuir la dimensionalidad de los datos y que no afecten la calidad de la clasificación supervisada cuando se usan los conjuntos de datos procesados por estos métodos.
2. Diseñar nuevos métodos para editar conjuntos de entrenamiento, que sean capaces de mejorar la efectividad en la clasificación supervisada.
3. Generar una propuesta para estimar la calidad de los resultados de la clasificación supervisada a partir de los conjuntos de entrenamiento para diferentes clasificadores a través del estudio de la capacidad de estimación de las medidas de la Teoría de los Conjuntos Aproximados.

Se formularon las siguientes **hipótesis** de investigación:

H1. Si se emplean los elementos de la Teoría de los Conjuntos Aproximados, se pueden desarrollar métodos para la selección de atributos relevantes y la edición de conjuntos de

entrenamiento, con el fin de reducir este conjunto y de mejorar la eficiencia computacional de la clasificación supervisada.

H2. Es posible caracterizar a priori los conjuntos de entrenamiento si se utilizan medidas de inferencia de la Teoría de los Conjuntos Aproximados para generar conocimiento sobre la precisión de clasificadores supervisados.

Las tareas de investigación trazadas son:

- El análisis de los métodos existentes de reducción de atributos, edición de prototipos y evaluación a priori de un conjunto de entrenamiento, así como la implementación de algunos de los métodos encontrados en la revisión bibliográfica.
- El desarrollo de algoritmos para la selección de atributos relevantes, ya sea a través de algoritmos evolutivos, como por la combinación de la Teoría de los Conjuntos Aproximados con búsqueda heurística.
- El desarrollo de algoritmos para la edición de conjuntos de entrenamiento a través de elementos de la Teoría de los Conjuntos Aproximados.
- Las correlaciones estadísticas establecidas entre los resultados obtenidos de las medidas de estimación de los Conjuntos Aproximados y la evaluación de los clasificadores, a través de conjuntos de datos publicados en el UCI repository for machine learning, para la generación de una propuesta que permita predecir la eficiencia de los clasificadores.
- La implementación de sistemas que posibiliten la generación automatizada de conjuntos de atributos relevantes, la edición de conjuntos de entrenamiento y el cálculo de las medidas de estimación de los Conjuntos Aproximados.
- La evaluación de los algoritmos y aplicaciones desarrolladas a partir de conjuntos de datos publicados en el UCI repository for machine learning y en conjuntos de datos de pronósticos del tiempo del Centro Meteorológico de la provincia de Camagüey.
- La aplicación de los métodos y herramientas desarrolladas en la implementación de un sistema que denominaremos PROMETEM, que realice de forma automatizada

pronósticos de temperatura; usando la información de los valores de las temperaturas, así como de variables asociadas a esta, en los siete años anteriores.

Por tales motivos se plantean las siguientes **preguntas de investigación**:

¿Cómo usar la Teoría de los Conjuntos Aproximados en el preprocesamiento de los datos, desde dos perspectivas: la creación de algoritmos tanto para la selección de atributos relevantes de un conjunto, como para la edición de los objetos de dicho conjunto?

¿Cómo usar la Teoría de los Conjuntos Aproximados en la caracterización a priori de los conjuntos de entrenamiento para la clasificación supervisada?

¿Si se combinan elementos de la Teoría de los Conjuntos Aproximados con técnicas de aprendizaje automatizado, se logran mejores resultados que los que se obtenían anteriormente para la predicción meteorológica, específicamente para las variables correspondientes a las temperaturas máximas y mínimas?

Entre los Métodos de trabajo científico utilizados se destacan los siguientes:

- Analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de las propuestas.
- Histórico-lógico y el dialéctico para el estudio crítico del los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.
- Hipotético-deductivo para la elaboración de las hipótesis de la investigación y para proponer nuevas líneas de trabajo a partir de los resultados parciales que se obtuvieron.
- Modelación para el desarrollo de los algoritmos.
- Sistémico para el desarrollo de los diferentes sistemas computacionales y lograr que los elementos que formen parte de la aplicación real sean un todo que funcione de manera armónica.

- Experimental para comprobar la utilidad de los resultados obtenidos a partir de las propuestas definidas y la comparación con otros métodos reportados.
- Matemáticos-estadísticos para la validación de los aportes fundamentales de la investigación.
- Análisis-síntesis e Inducción-deducción como vía de constatación teórica a lo largo de la tesis.
- Coloquial para la presentación y discusión de los resultados en sesiones científicas.

La **novedad científica** del presente trabajo se resume en los puntos siguientes:

1. Se proponen algoritmos para la selección de atributos relevantes. Uno de ellos, construye reductos a través de un algoritmo de distribución marginal univariado. El otro obtiene un reducto a través de la Teoría de los Conjuntos Aproximados, donde se construyen funciones heurísticas para determinar relevancia de atributos. En este último, se siguen criterios de entropía, ganancia y conceptos de las aproximaciones de los Conjuntos Aproximados.
2. Se proponen nuevos algoritmos de edición de conjuntos de entrenamiento, basados en los conceptos de aproximación inferior y función de pertenencia aproximada de la Teoría de los Conjuntos Aproximados. De esta manera, no solo se seleccionan objetos, sino que se construyen otros a partir de la información que se obtiene de ellos sobre la pertenencia al resto de las clases.
3. Se desarrolla una propuesta para caracterizar a priori los conjuntos de entrenamiento, donde se toma la información de las medidas de estimación de los Conjuntos Aproximados en la construcción de atributos predictores para generar conocimiento a partir de ellas. De esta manera, se da una estimación de la aplicabilidad y el desempeño de clasificadores supervisados para estos conjuntos de entrenamiento.

Por su parte, el **valor práctico** está encaminado a la aplicación de estos nuevos algoritmos en un problema real del Centro Meteorológico de Camagüey, en la automatización del pronóstico de las temperaturas máximas y mínimas, al abordar este de forma integral desde el punto de vista de caracterizar a priori el conjunto de entrenamiento, seleccionar sus

atributos relevantes y editar el conjunto de objetos que lo describe; de esta forma se logra un preprocesamiento de los datos con vistas a mejorar los resultados en el aprendizaje inductivo.

Además, los desarrollos teóricos alcanzados en este trabajo han dado como resultado cuatro aplicaciones informáticas de alto valor práctico:

- El sistema RoughSetsReduct 2.0 que posibilita la selección automatizada de atributos relevantes.
- El sistema EditCE 1.0 que permite de forma automatizada editar conjuntos de entrenamiento.
- El sistema MIRST 1.0 con el que es posible el cálculo de las medidas de estimación de los Conjuntos Aproximados.
- El sistema PROMETEM 1.0 para el pronóstico de los valores diarios de las temperaturas máximas y mínimas.

Estos sistemas han sido empleados con éxito no solo en la solución de problemas reales como el del Centro Meteorológico de Camagüey para la predicción meteorológica, en el preprocesamiento de los conjuntos de datos utilizados, sino para fines docentes y de investigación en Ciencias de la Computación e Ingeniería Informática, por su grado de generalidad.

La **tesis** está **estructurada** en tres capítulos:

El Capítulo 1 trata el problema del aprendizaje, se hace énfasis en el aprendizaje a partir de ejemplos y en algunos métodos existentes. Se describen además los aspectos fundamentales de la Teoría de los Conjuntos Aproximados. Se realiza un análisis crítico en la selección de atributos relevantes, la edición de conjuntos de entrenamiento, y la caracterización a priori de los conjuntos de entrenamiento, enfatizando en aquellos algoritmos que resultaron más interesantes. Además, se dan los elementos fundamentales acerca del problema real de predicción de las temperaturas máximas y mínimas en el Centro Meteorológico de Camagüey, como uno de los problemas reales donde es necesario el preprocesamiento de los datos.

En el Capítulo 2 se proponen dos métodos para la selección de atributos relevantes, uno de ellos combina elementos de la computación evolutiva; el otro, a través de heurísticas logra construir un conjunto de atributos relevantes. Además se proponen dos nuevos algoritmos de edición: Edit1RS y Edit3RS basados en la RST, los cuales se diseñan e implementan en la realización de esta tesis. Los resultados de los métodos propuestos se validan en este capítulo usando conjuntos de datos internacionales. Se realiza un estudio para caracterizar a priori los conjuntos de entrenamiento, basado en las medidas de estimación de los Conjuntos Aproximados; se proponen nuevas medidas en este sentido. Se establecen correlaciones estadísticas de los resultados de las medidas de estimación y los clasificadores Perceptron multicapa (MLP), k-Vecinos más Cercanos (k-NN) y C4.5, y se genera conocimiento a partir de las medidas.

En el Capítulo 3 se validan los resultados teóricos de la investigación en el preprocesamiento de los datos del problema de predicción de las temperaturas máximas y mínimas para las seis estaciones del Centro Meteorológico de Camagüey.

Este documento culmina con las conclusiones, recomendaciones, bibliografía, producción científica de la autora sobre el tema de la tesis y los anexos.

Capítulo 1

Acerca del aprendizaje automatizado, la selección de atributos y la edición de conjuntos de entrenamiento

1 ACERCA DEL APRENDIZAJE AUTOMATIZADO, LA SELECCIÓN DE ATRIBUTOS Y LA EDICIÓN DE CONJUNTOS DE ENTRENAMIENTO

Al igual que muchos problemas de la Inteligencia Artificial (IA), el aprendizaje atrae la atención de los científicos de la computación y otras disciplinas. En particular, el aprendizaje científico recibe una atención considerable (Rich and Knight 1994).

¿Qué es el aprendizaje?

Una de las críticas que se oyen más a menudo respecto a la IA, es que las máquinas no se pueden considerar inteligentes hasta que no sean capaces de aprender a hacer cosas nuevas y a adaptarse a las nuevas situaciones, en lugar de limitarse a hacer aquellas actividades para las que fueron programadas. En vez de preguntarse si una computadora es capaz de "aprender", resulta mucho más clarificador intentar describir a qué actividades se refiere exactamente cuando se dice "aprender " y cuáles mecanismos se pueden utilizar para llevar a cabo dichas actividades. Simon en 1983 (Simon 1983) se refiere al aprendizaje como cambios en el sistema para desarrollar tareas a partir de las mismas condiciones de un modo más eficiente y eficaz cada vez.

La mayoría de los programas de IA confían principalmente en el conocimiento como fuente de recursos. El conocimiento se adquiere generalmente a través de la experiencia. La adquisición del conocimiento en sí misma incluye muchas actividades diferentes. El simple almacenamiento de información computada, también conocido como aprendizaje memorístico, constituye la actividad de aprendizaje más básica (Kelly Christian 2001).

El aprendizaje a partir de ejemplos puede requerir o no un profesor que ayude a clasificar corrigiendo cuando uno se equivoca. Un caso típico es el problema de clasificación supervisada, donde se aprende a clasificar sin haber recibido reglas explícitas (Rich and Knight 1994).

La clasificación supervisada consiste en el proceso de asignar a una entrada concreta, el nombre de una clase a la que pertenece. Las clases entre las que puede elegir el procedimiento de clasificación supervisada se pueden describir de gran cantidad de formas. La clasificación supervisada constituye una parte importante de muchas de las tareas de resolución de problemas y en su forma más simple, se presenta como una tarea de

reconocimiento. Antes de que se pueda hacer este tipo de clasificación, se deben definir las clases que utilizará.

1.1 Aprendizaje Automatizado

Los grandes avances en las tecnologías de la información han provocado un aumento sin precedentes tanto en el volumen de datos como en su flujo. Códigos de barra, tarjetas electrónicas, sensores remotos, transacciones bancarias, satélites espaciales o el descifrado del código genético humano son ejemplos de la necesidad de filtrar, analizar e interpretar volúmenes de datos que se miden en terabytes.

Desde hace más de dos décadas se vienen desarrollando y utilizando complejos algoritmos para la extracción de patrones útiles en grandes conjuntos de datos. Gran parte de esta información representa transacciones o situaciones que se han producido, siendo útil no sólo para explicar el pasado, sino para entender el presente y predecir la información futura. En muchas ocasiones, el método tradicional de convertir los datos en conocimiento consiste en un análisis e interpretación realizada de forma manual por especialistas en la materia estudiada. Esta forma de actuar es lenta, cara y altamente subjetiva. De hecho, la enorme cantidad de datos desborda la capacidad humana de comprenderlos y el análisis manual hace que las decisiones se tomen según la intuición de los especialistas.

A finales de la década de los 80, la creciente necesidad de automatizar todo este proceso inductivo abre una línea de investigación para el análisis inteligente de datos, impulsando las investigaciones en aprendizaje automatizado. El Aprendizaje Automatizado es el área de la Inteligencia Artificial que se ocupa de desarrollar técnicas capaces de aprender, es decir, extraer de forma automatizada conocimiento subyacente en la información. Constituye, junto con la estadística, el corazón del análisis inteligente de los datos (Ruiz 2006).

El principio seguido en el aprendizaje automatizado es que la máquina genera un modelo a partir de ejemplos y lo usa para resolver el problema. Uno de los modelos de aprendizaje más estudiado es el aprendizaje inductivo, que engloba todas aquellas técnicas que aplican inferencias inductivas sobre un conjunto de datos para adquirir conocimiento inherente a ellos.

1.1.1 Métodos de aprendizaje automatizado a partir de ejemplos

Los métodos de aprendizaje y clasificación pueden ser organizados, atendiendo a su naturaleza, en métodos estadísticos, modelos o algoritmos matemáticos para el reconocimiento de patrones, estrategias basadas en árboles de decisión, sistemas basados en el conocimiento, entre otras (Piñero 2005). Existen varias formas de representar el conocimiento, las reglas constituyen una de las más usadas pues su modo de representación es el del razonamiento lógico de los seres humanos. En el desarrollo de los sistemas expertos resulta engorroso el mecanismo de aprendizaje si se necesita de los especialistas para poder generar las reglas. Con el surgimiento del aprendizaje automatizado esta generación de reglas se realiza automáticamente mediante algoritmos.

Para construir un clasificador usando un aprendizaje a partir de datos, los ejemplos son pares de la forma (*A, c*), donde *A* es un conjunto de atributos que describen el objeto y *c* es la clase del objeto. La construcción del clasificador significa encontrar una función *g* tal que $c=g(A)$. El algoritmo de aprendizaje genera una función *h* que aproxima *g* (Klose 2003). Los métodos para construir *h* se dividen en inductivos y perezosos. Ejemplos de métodos inductivos son el Perceptron multicapa (Rosemblatt 1962) y el algoritmo C4.5 (Quinlan 1993), mientras que el k-NN (Cover and Hart 1967) es un típico método de aprendizaje perezoso (Freeman and Skapura 1991; Mitchell 1997; Witten and Frank 2005a).

Los Conjuntos Aproximados por su parte, se han usado para resolver varias tareas en el aprendizaje automatizado, entre ellas, la generación de reglas (Koczkodaj 1998; Ohrn, Komorowski et al. 1998; Gomolinska 2005), la selección de atributos (Kohavi and Frasca 1994; Pawlak 1995b; Yao and Wong 1999; Dunstsh and Gunter 2000), y en el aprendizaje perezoso se han empleado también para la construcción de casos (Pal and Mitra 2001). Por lo que desde estos puntos de vista se puede ver su amplia aplicación en la problemática del aprendizaje a partir de ejemplos.

Redes Neuronales Artificiales

Existen varios modelos de Redes Neuronales Artificiales que se emplean en la solución de problemas de clasificación supervisada, uno de los más referenciados es el Perceptron

multicapa (MLP) (Rosemblatt 1962; Witten and Frank 2005a). El MLP es reconocido como la mejor red neuronal para solucionar un problema de clasificación supervisada a partir de ejemplos, no obstante, se le señala que se comporta como una caja negra y no facilita la interpretación de los resultados (Borgelt and Kruse 2003).

En la actualidad se han desarrollado múltiples aplicaciones en diversas ramas donde se han utilizado las Redes Neuronales Artificiales para su solución, tales como: predicción de heladas en la estación de Río Cuarto, Córdoba, Argentina, en el año 2005 (Ovando, Bocco et al. 2005); construcción de un sistema automatizado para el riesgo de mortalidad hospitalaria en una unidad de cuidados intensivos, en el año 2005 (Cabello, Castello et al. 2005); predicción de lluvia-escorrentía en tiempo real en el entorno del Sistema Automático de Información Hidrológica de la Cuenca Sur de España (SAIH-SUR), en el año 2006 (Rueda 2006); construcción de un sistema automatizado de valoración de viviendas en la ciudad de Albacete, España, en el año 2006 (García, Gámez et al. 2006); entre otras (Borracci and Arribalzaga 2005; Milone 2005; Zúñiga and Jordán 2005; Calderón and Jara 2006; González and Cases 2006; Romero, Machado et al. 2006).

Árboles de decisión

Una estrategia pionera en este campo es el análisis CHAID (Chi-square Automatic Interaction Detector) cuya extensión ha dado origen a otros métodos (Piñero, Grau et al. 2002). En general la inducción de los árboles de decisión está basada en algoritmos de aprendizaje discriminativo por medio de particionamientos recursivos. Los árboles de decisión son una de las formas más sencillas de representación del conocimiento adquirido (Moshkov 2005). Entre los algoritmos basados en árboles de decisión están el ID3 (Mitchell 1997), que solo permite el tratamiento de datos simbólicos, y sus extensiones C4.5 (Quinlan 1993) y C5.0 (Quinlan 2000) que posibilitan además el tratamiento de datos numéricos.

El algoritmo C4.5 ha sido uno de los sistemas clasificadores más referenciados en la literatura, principalmente debido a su extremada robustez en un gran número de dominios y su bajo costo computacional. Este método trata eficazmente los valores desconocidos calculando la ganancia de información para los valores presentes y maneja los atributos

continuos. Para un conjunto de datos con *m* objetos y *n* atributos, el costo medio de construcción del árbol es de $\theta(mn\log_2 m)$, mientras que la complejidad del proceso de poda es de $\theta(m(\log_2 m)^2)$ (Ruiz 2006). Recientemente, múltiples problemas han sido resueltos a través de los árboles de decisión, específicamente usando el algoritmo C4.5, entre ellos: Toma de decisiones para el tratamiento del cáncer, según síntomas e historias clínicas de los pacientes, en el año 2001 (Elwyn, Edwards et al. 2001); toma de decisiones en la planificación del uso de la tierra en los Llanos Orientales de Colombia, para evitar su degradación, en el año 2003 (Quintero, Amezquita et al. 2003); sistema para la extracción de conocimiento en Bioinformática, en el año 2004 (Ferri, Hernández et al. 2004); toma de decisiones en el dolor precordial como síntoma de las enfermedades del corazón, en el año 2006 (Sánchez 2006); entre otras.

Razonamiento basado en casos

Los sistemas basados en casos basan sus potencialidades para el aprendizaje en los mecanismos de detección de similitudes entre casos y también en la propiedad de incorporar de forma natural nuevo conocimiento mejorando gradualmente su funcionamiento (Gutiérrez 2003). Su principal limitación radica en la capacidad de representación de los valores de los atributos, aspecto que dificulta su aplicación en situaciones que requieren representaciones de casos complejas como las que se presentan cuando existe un alto grado de interrelación entre los atributos (Piñero 2005). El método de los k-Vecinos más Cercanos (k-NN) es un ejemplo de razonamiento basado en casos.

Los fundamentos de la clasificación supervisada por vecindad, que se basan en seleccionar el ejemplo más similar al problema de acuerdo a una función de distancia (o semejanza), fueron establecidos por E. Fix y J. L. Hodges (Fix and Hodges 1951) a principio de los años 50. Sin embargo, no fue hasta 1967 cuando T. M. Cover y P. E. Hart (Cover and Hart 1967) enuncian formalmente la Regla del Vecino más Cercano y la desarrollan como herramienta de clasificación supervisada de patrones. Desde entonces, este algoritmo k-NN se ha convertido en uno de los métodos de clasificación supervisada más usados (Cover and Hart 1967; Aha, Kibler et al. 1991; Dasarathy, Sanchez et al. 2000). Algunos ejemplos de aplicaciones actuales que lo han llevado a ser un clasificador muy empleado son:

predicción de la visibilidad e intensidad del viento en terminales aeroportuarias, en el año 2002 (Riordan and Hansen 2002); evaluación de la calidad interna de la madera de los árboles en la producción de "madera de piceas" en Noruega, en el año 2003 (Malinen, Maltamo et al. 2003); clasificación automática de imágenes de muestras de endometrio humano, en el año 2006 (Rojas 2006); entre otras (García, Vidal et al. 2004; Sordo 2006).

1.2 Conjuntos Aproximados

La Teoría de Conjuntos Aproximados (Rough Sets Theory) fue introducida por Z. Pawlak en 1982 (Pawlak 1982). Se basa en aproximar cualquier concepto, un subconjunto duro del dominio como por ejemplo, una clase en un problema de clasificación supervisada, por un par de conjuntos exactos, llamados aproximación inferior y aproximación superior del concepto. Con esta teoría es posible tratar tanto datos cuantitativos como cualitativos, y no se requiere eliminar las inconsistencias previas al análisis; respecto a la información de salida puede ser usada para determinar la relevancia de los atributos, generar las relaciones entre ellos (en forma de reglas), entre otras (Choubey 1996; Chouchoulas and Shen 1999; Greco, Inuiguchi et al. 2003; Miao and Hou 2003; Midelfart, Komorowski et al. 2003; Piñero, Arco et al. 2003; Tsumoto 2003; Zhao, Zhang et al. 2003; Grzymala-Busse and Siddhaye 2004; Sugihara and Tanaka 2006). La inconsistencia describe una situación en la cual hay dos o más valores en conflicto para ser asignados a una variable (Bosc and Prade 1993; Parsons 1996).

Sobre los Conjuntos Aproximados se han manifestado diversos autores, los cuales ven esta teoría como la mejor herramienta para modelar la incertidumbre cuando esta se manifiesta en forma de inconsistencia, y como una nueva dirección en el desarrollo de teorías sobre la información incompleta (Grzymala-Busse 1994; Orlowska 1998; Pal and Skowron 1999; Greco 2001; Pal 2002; Grabowski 2003; Skowron and Peters 2003).

En este epígrafe se describirán los conceptos fundamentales de los Conjuntos Aproximados, tanto para el caso clásico como para el enfoque basado en relaciones de similitud.

Principales definiciones de la Teoría de los Conjuntos Aproximados

La filosofía de los conjuntos aproximados se basa en la suposición de que con todo objeto *x* de un universo *U* está asociada una cierta cantidad de información (datos y conocimiento), expresado por medio de algunos atributos que describen el objeto (Komorowski and Pawlak 1999; Bazan, Son et al. 2003).

Diversos modelos computacionales operan sobre colecciones de datos. En cada caso esta colección tiene sus características, sobre todo organizativas, y recibe una denominación particular. Por ejemplo, para un Gestor de Bases de Datos esa colección es una base de datos, para una Red Neuronal Artificial es un conjunto de entrenamiento. En el caso de la Teoría de los Conjuntos Aproximados la estructura de información básica es el Sistema de Información.

Definición 1. Sistema de Información y sistema de decisión

Sea un conjunto de atributos $A=\{a_1, a_2,\ldots,a_n\}$ y un conjunto *U* no vacío llamado universo de ejemplos (objetos, entidades, situaciones o estados) descritos usando los atributos a_i; al par *(U,A)* se le denomina Sistema de información (Komorowski and Pawlak 1999)[1]. Si a cada elemento de *U* se le agrega un nuevo atributo *d* llamado decisión, indicando la decisión tomada en ese estado o situación, entonces se obtiene un Sistema de decisión $(U,A\cup\{d\})$, donde $d\notin A$.

Definición 2. Función de información

A cada atributo a_i se le asocia un dominio v_i. Se tiene una función $f: UxA\rightarrow V$, $V=\{v_1,v_2,\ldots,v_p\}$ tal que $f(x,a_i)\in v_j$ para cada $a_i\in A$, $x\in U$, llamada función de información (Komorowski and Pawlak 1999).

El atributo de decisión *d* induce una partición del universo *U* de objetos. Sea el conjunto de enteros $\{1,\ldots,l\}$, $X_i=\{x\in U: d(x)=i\}$, entonces $\{X_1,\ldots,X_l\}$ es una colección de clases de equivalencias, llamadas clases de decisión, donde dos objetos pertenecen a la misma clase si ellos tienen el mismo valor para el atributo decisión.

[1] Esta definición es independiente a la definición de Sistema de Información de Shannon

Se dice que un atributo $a_i \in A$ separa o distingue un objeto x de otro y, y se escribe *Separa* (a_i, x, y), si y solo si se cumple:

$$f(x, a_i) \neq f(y, a_i) \tag{1.1}$$

La relación de separabilidad se basa en la comparación de los valores de un atributo, para lo cual se ha usado la igualdad (o desigualdad) estricta. Sin embargo, es posible usar una condición de comparación menos estricta definida de esta forma:

$$Separa\ (a_i, x, y) \Leftrightarrow |f(x, a_i) - f(y, a_i)| > \varepsilon \tag{1.2}$$

Definición 3. Relación de inseparabilidad

A cada subconjunto de atributos B de A $B \subseteq A$ está asociada una relación binaria de inseparabilidad denotada por R, la cual es el conjunto de pares de objetos que son inseparables uno de otros por esa relación (Komorowski and Pawlak 1999).

$$R = \{(x, y) \in UxU : f(x, a_i) = f(y, a_i) \forall a_i \in B\} \tag{1.3}$$

Una relación de inseparabilidad (indiscernibility relation) que sea definida a partir de formar subconjuntos de elementos de U que tienen igual valor para un subconjunto de atributos B de A, $B \subseteq A$, es una relación de equivalencia.

Los conceptos básicos de la RST son las aproximaciones inferiores y superiores de un subconjunto $X \subseteq U$. Estos conceptos fueron originalmente introducidos con referencia a una relación de inseparabilidad R. Sea R una relación binaria definida sobre U la cual representa la inseparabilidad, se dice que $R(x)$ significa el conjunto de objetos los cuales son inseparables de x. Así, $R(x) = \{y \in U : yRx\}$. En la RST clásica, R es definida como una relación de equivalencia; es decir, es una relación binaria $R \subseteq UxU$ que es reflexiva, simétrica y transitiva. R induce una partición de U en clases de equivalencia correspondiente a $R(x)$, $x \in U$.

Este enfoque clásico de RST es extendido mediante la aceptación que objetos que no son inseparables pero sí suficientemente cercanos o similares puedan ser agrupados en la misma clase (Slowinski and Vanderpooten 1997). El objetivo es construir una relación R' a partir de la relación de inseparabilidad R pero flexibilizando las condiciones originales para la

inseparabilidad. Esta flexibilización puede ser realizada de múltiples formas, así como pueden ser dadas varias definiciones posibles de similitud. Existen varias funciones de comparación de atributos (funciones de similitud), las cuales están asociadas al tipo del atributo que se compara (Wilson and Martínez 1997; García 2003). Sin embargo, la relación *R'* debe satisfacer algunos requerimientos mínimos. Si *R* es una relación de inseparabilidad definida en *U*, *R'* es una relación de similitud extendida de *R* si y solo si $\forall x \in U, \quad R(x) \subseteq R'(x)$ y $\forall x \in U$, $\forall y \in R'(x)$, $R(y) \subseteq R'(x)$, donde *R'*(*x*) es la clase de similitud de *x*, es decir, $R'(x)=\{y \in U: yR'x\}$. *R'* es reflexiva, cualquier clase de similitud puede ser vista como un agrupamiento de clases de inseparabilidad y *R'* induce un cubrimiento de *U* (Skowron and Stepaniuk 1992). Esto muestra que un objeto puede pertenecer a diferentes clases de similitud simultáneamente, lo que significa que el cubrimiento inducido por *R'* sobre *U* no es necesariamente una partición.

La aproximación de un conjunto $X \subseteq U$, usando una relación de inseparabilidad *R*, ha sido inducida como un par de conjuntos llamados aproximaciones *R*-inferior y *R*-superior de *X*. Se considera en esta tesis una definición de aproximaciones más general, la cual maneja cualquier relación reflexiva *R'*. Las aproximaciones *R'*-inferior ($R'_*(X)$) y *R'*-superior ($R'^*(X)$) de *X* están definidas respectivamente como se muestra en las expresiones (1.4) y (1.5).

$$R'_*(X) = \{x \in X : R'(x) \subseteq X\} \tag{1.4}$$

$$R'^*(X) = \bigcup_{x \in X} R'(x) \tag{1.5}$$

Teniendo en cuenta las expresiones definidas en 1.4 y 1.5, se define la región límite de *X* para la relación *R'* (Deogun 1995):

$$BN_B(X) = R'^*(X) - R'_*(X) \tag{1.6}$$

Si el conjunto BN_B es vacío entonces el conjunto *X* es exacto respecto a la relación *R'*. En caso contrario, $BN_B(X) \neq \phi$, el conjunto *X* es inexacto o aproximado con respecto a *R'*.

El uso de relaciones de similitud ofrece mayores posibilidades para la construcción de las aproximaciones; sin embargo, se tiene que pagar por esta mayor flexibilidad, pues es más

difícil desde el punto de vista computacional buscar las aproximaciones relevantes en este espacio mayor (Pal and Skowron 1999).

Existen varias funciones de comparación de atributos (funciones de similitud), las cuales están asociadas al tipo de los atributos. Algunas de estas funciones de similitud se describen en (Wilson and Martínez 1997; García 2003).

Usando las aproximaciones inferior y superior de un concepto X se definen tres regiones para caracterizar el espacio de aproximación: la región positiva que es la aproximación R'-inferior, la región límite que es el conjunto BN_B y la región negativa ($NEG(X)$) que es la diferencia entre el universo y la aproximación R'-superior. Los conjuntos $R'_*(X)$ (denotado también como $POS(X)$), $R'^*(X)$, $BN_B(X)$ *y* $NEG(X)$ son las nociones principales de la Teoría de Conjuntos Aproximados.

Un aspecto importante en la Teoría de los Conjuntos Aproximados es la reducción de atributos basada en el concepto de reductos. Un reducto es un conjunto reducido de atributos que preserva la partición del universo (Komorowski and Pawlak 1999; Zhong, Dong et al. 2001). El uso de reductos en la selección y reducción de atributos ha sido ampliamente estudiado (Kohavi and Frasca 1994; Carlin 1998; Komorowski and Pawlak 1999; Pal and Skowron 1999; Ahn 2000; Lazo, Ruiz et al. 2001; Zhong, Dong et al. 2001; Santiesteban and Pons 2003; Caballero 2005).

Medidas de inferencia clásicas de la Teoría de los Conjuntos Aproximados

La Teoría de los Conjuntos Aproximados ofrece algunas medidas para analizar los sistemas de información (Skowron 1999; Arco, Bello et al. 2006). A continuación se muestran las principales. En las expresiones 1.7 a la 1.10 se emplean las aproximaciones R'-inferior ($R'_*(X)$) y R'-superior ($R'^*(X)$) de X, las cuales están definidas en las expresiones 1.4 y 1.5 respectivamente.

Precisión de la aproximación. Un conjunto aproximado X puede ser caracterizado numéricamente por el coeficiente llamado precisión de la aproximación, donde $|X|$ denota la cardinalidad de X, $X \neq \phi$. Observe la expresión (1.7).

$$\alpha(X) = \frac{\left|R'_*(X)\right|}{\left|R'^*(X)\right|} \tag{1.7}$$

Obviamente, $0 \leq \alpha(X) \leq 1$. Si $\alpha(X)=1$, X es duro (exacto), si $\alpha(X)<1$, X es aproximado (vago, inexacto), siempre respecto al conjunto de atributos considerado (Skowron 1999).

Calidad de la aproximación. El coeficiente siguiente

$$\gamma(X) = \frac{\left|R'_*(X)\right|}{\left|X\right|} \tag{1.8}$$

expresa la proporción de objetos que pueden ser correctamente clasificados en la clase X. Además, $0 \leq \alpha(X) \leq \gamma(X) \leq 1$, y $\gamma(X)=0$ si y solo si $\alpha(X)=0$, mientras $\gamma(X)=1$ si y solo si $\alpha(X)=1$ (Skowron 1999).

Considerando que $X_1,\ldots,X_l$ son las clases del sistema de decisión, se define la medida:

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

$$\Gamma(DS) = \frac{\sum_{i=1}^{l}\left|R'_*(X_i)\right|}{\left|U\right|} \tag{1.9}$$

La medida calidad de la clasificación expresa la proporción de objetos que pueden estar correctamente clasificados en el sistema. Si ese coeficiente es igual a 1, entonces el sistema de decisión es consistente, en otro caso es inconsistente (Skowron 1999).

Función de pertenencia aproximada. Esta función cuantifica el grado de solapamiento relativo entre $R'(x)$ (clase de similitud de x) y la clase a la cual el objeto x pertenece. Se define como sigue:

$$\mu_X(x) = \frac{\left|X \cap R'(x)\right|}{\left|R'(x)\right|} \tag{1.10}$$

La función de pertenencia aproximada puede ser interpretada como una estimación basada en frecuencias de $Pr(x \in X \mid x, R'(x))$, es decir, la probabilidad condicional de que el objeto x pertenezca al conjunto X (Grabowski 2003).

En el epígrafe 2.1 de esta tesis, como resultado de la investigación, se describe un método con dos heurísticas, cuyo objetivo es seleccionar un conjunto de atributos relevantes a través de los Conjuntos Aproximados. Otra aplicación de los Conjuntos Aproximados es en la edición de conjuntos de entrenamiento, tal es el caso de los métodos que se describen en el epígrafe 2.2. Además, se definen nuevas medidas de inferencia de los Conjuntos Aproximados en el epígrafe 2.4.

1.3 Preprocesamiento de los datos

La utilidad del conocimiento extraído a partir de los datos mediante los métodos de aprendizaje automatizado depende en gran medida de la calidad de esos datos. Existe una fase de preparación de los datos previa a su análisis. Pyle y colaboradores en (Pyle 1999) indican que el propósito fundamental de esta fase es el de manipular y transformar los datos en bruto, de manera que la información contenida en el conjunto de datos pueda ser descubierta, o más fácilmente accesible (Ruiz 2006).

Debido a que en muchas ocasiones los datos provienen de diferentes fuentes, pueden contener valores incompletos (ausencia de información), con ruido (error aleatorio o variación en el valor de un atributo, debido normalmente a errores en la medida del mismo) e inconsistentes, y esto puede conducir a la extracción de patrones poco útiles. Además, en el preprocesamiento es posible reducir el conjunto de datos (selección de atributos y de objetos), mejorando la eficiencia del proceso de descubrimiento de conocimiento posterior. También existe la posibilidad de recuperar información incompleta, eliminar ruidos, resolver conflictos, etcétera, generando un conjunto de datos de calidad, que conduciría a mejores patrones (Ruiz 2006).

La preparación o preprocesamiento de datos engloba a todas aquellas técnicas de análisis de datos que permiten mejorar la calidad de un conjunto de ellos, de modo que los métodos de extracción de conocimiento puedan obtener mayor y mejor información (mejor desempeño en la clasificación supervisada, reglas con más completitud, entre otros) (Zhang, Zhang et al. 2003). Los procesos que se incluyen en esta fase se pueden resumir en: recopilación de datos, limpieza, transformación y reducción, y no siempre se aplican en un mismo orden (Ruiz 2006) (ver figura 1.1).

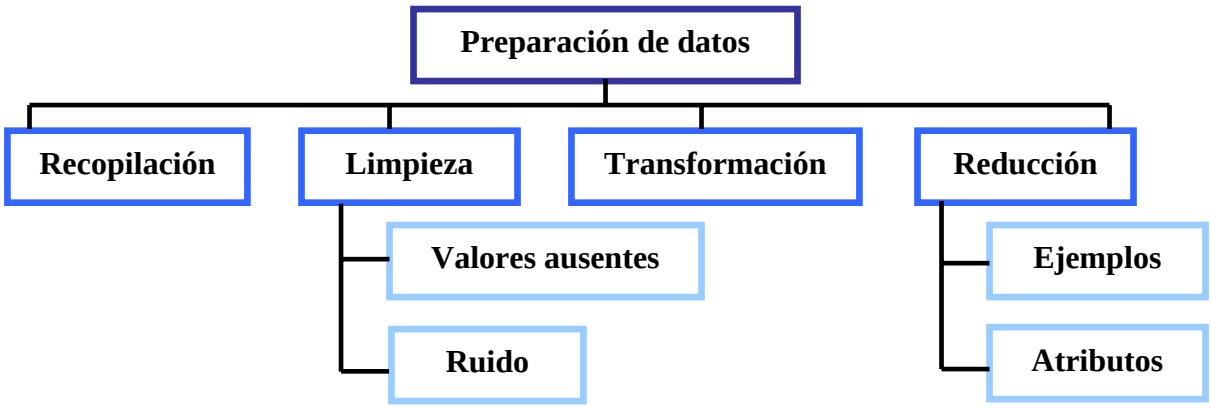

Figura 1.1 Fase de preparación o preprocesamiento de los datos

Para poder comenzar a analizar y extraer algo útil en los datos es preciso, en primer lugar, disponer de ellos. Esto en algunos casos puede parecer trivial, a partir de un simple archivo de datos, sin embargo en otros, es una tarea muy compleja dado por problemas de representación, de codificación e integración de diferentes fuentes para crear información homogénea, este es el llamado proceso de recopilación. En la fase de limpieza se resuelven conflictos entre datos, y se comprueban problemas de ruido, valores ausentes, entre otros (Ruiz 2006).

En ocasiones, la forma en que viene dada la información originalmente no es la más adecuada para adquirir conocimiento a partir de ella. En esas situaciones se hace necesaria la aplicación de algún tipo de transformación para adecuar los datos al posterior proceso de aprendizaje, como por ejemplo normalización o cambio de escala, discretización, generalización o extracción de atributos. Esta última transformación está estrechamente relacionada con la selección de características, detallada más adelante y consiste en extraer conjuntos de atributos a partir de combinaciones de los originales. Es conveniente, aplicar técnicas de reducción al conjunto de datos, orientadas fundamentalmente hacia dos objetivos: técnicas de selección de atributos (eliminar aquellos atributos que no sean relevantes para la información inherente al conjunto de datos) y técnicas de edición (reducción del número de objetos o ejemplos para mejorar el desempeño de los clasificadores con menos costo computacional). En la figura 1.1 se pueden observar los dos tipos de reducción.

1.4 Selección de atributos relevantes

La selección de atributos es un problema que aparece en diferentes tareas asociadas con el análisis de datos (Xu and Setiono 2003; Yu and Liu 2004a; Yu and Liu 2004b; Ruiz, Aguilar et al. 2005), y tiene especial relevancia en las áreas de Reconocimiento de Patrones y Aprendizaje Automatizado (Ruiz, Aguilar-Ruiz et al. 2004; Liu and Yu 2005). El problema de selección de atributos consiste en encontrar el subconjunto de atributos que mejor describe los objetos del dominio; usualmente este subconjunto se encuentra maximizando o minimizando una función objetivo. Los procedimientos de selección de atributos constan de dos componentes principales: la función de evaluación (una función heurística) y el método de generación de subconjuntos. Existen diferentes enfoques y técnicas para seleccionar atributos relevantes, tales como el enfoque lógico combinatorio, el estadístico, las técnicas de optimización de colonias de hormigas, las de computación evolutiva, las basadas en la Teoría de los Conjuntos Aproximados, entre otros. A continuación se mencionan brevemente algoritmos que siguen algunos de estos enfoques y técnicas.

Algoritmos basados en el Enfoque Lógico Combinatorio

Dentro de este enfoque se enmarcan múltiples algoritmos de Reconocimiento de Patrones que seleccionan conjuntos representativos de atributos, tales como: el algoritmo de Yablonskii (Cheguis and Yablonskii 1958; Dimitriev, Zhuravlev et al. 1966), los algoritmos BT y TB (Ruiz, Aguila et al. 1985), el algoritmo LEX (Santiesteban and Pons 2003), entre otros. Estos algoritmos están basados en el concepto de testor típico (subconjunto mínimo de atributos que permite diferenciar objetos de clases distintas) (Lazo, Ruiz et al. 2001; Carrasco, Ruíz et al. 2004). La limitante que tienen es que calculan exhaustivamente todos los posibles subconjuntos que cumplen esta propiedad, por lo que son en extremo engorrosos y a su vez, mientras más grande sea la dimensionalidad de la matriz, se incrementa el costo computacional de estos métodos.

Algoritmos de Optimización de Colonias de Hormigas (OCH)

Los OCH reproducen el comportamiento de las hormigas reales en una colonia artificial. El cálculo de reductos puede ser reformulado en técnicas de colonias de hormigas de forma

factible si se representa cada atributo en un nodo del grafo y los arcos entre ellos denotan la posibilidad de optar por el siguiente atributo. La búsqueda del reducto es entonces un recorrido de una hormiga por el grafo donde sea visitada la cantidad mínima de nodos que satisfaga el criterio de parada. El modelo OCH utiliza varios parámetros, sería necesario estudiar cuáles son los valores adecuados para estos, al aplicar el modelo en la selección de atributos.

En (Jensen and Qiang 2003) se utiliza como función de evaluación heurística la medida de dependencia entre los atributos de la teoría de los conjuntos aproximados y se implementa de esta forma un algoritmo OCH para el cálculo de reductos. Se han reportado otros métodos de selección de atributos a través de los algoritmos de Optimización de Colonias de Hormigas que muestran resultados positivos (Bello, Nowe et al. 2005a; Bello, Nowe et al. 2005b). Sin embargo, como este es un modelo nuevo, se han publicado pocos trabajos sobre él, por lo que es necesaria más experimentación para poder tener una valoración.

Algoritmos Evolutivos

Los Algoritmos Genéticos (AG) son métodos de búsqueda y optimización sobre un espacio de soluciones potenciales, basados en el principio de la selección natural y la evolución de poblaciones. El espacio de soluciones potenciales o población es iterativamente refinado para optimizar la medida de puntaje de la población; esta medida se define a través de una función de puntaje (fitness function) de los individuos que se interpreta también como la habilidad de sobrevivir en el ambiente de dichos individuos. En (Wróblewski 1995) se presentan tres métodos para encontrar reductos cortos, a través de algoritmos genéticos. Otros métodos son los reportados en (Sánchez, Lazo et al. 1999; Wróblewski 2000; Martínez, Sánchez et al. 2002; Sánchez, Barandela et al. 2003; Santos and José 2003; Guevara 2004).

Los algoritmos evolutivos, de manera general, son eficientes en la optimización e imponen pocas restricciones a los modelos (Herrera 2004); sin embargo, los AG clásicos presentan dificultades en la optimización de funciones con interacciones no lineales entre las variables (Mühlenbein, Mahnig et al. 1999; Pelikan, Golberg et al. 2002; Cotta and Moscato 2004). O sea, por su naturaleza están preparados para el trabajo con variables

independientes; la aplicación de los operadores tradicionales de cruzamiento y mutación no incide en esta interrelación, lo que puede provocar dificultades en el aprendizaje en problemas donde existan interrelaciones marcadas entre los atributos.

Una adaptación del esquema de los Algoritmos Genéticos son los Algoritmos de Estimación de Distribuciones (EDA) (Mühlenbein, Mahnig et al. 1999). Los EDAs son algoritmos heurísticos de optimización basados en poblaciones que evolucionan. En los EDAs la evolución de las poblaciones no se lleva a cabo por medio de los operadores de cruce y mutación; en lugar de ello la nueva población de individuos se muestrea a partir de una distribución de probabilidad, la cual es estimada de la base de datos que contiene al conjunto de individuos seleccionados de la generación anterior. En (Saeys, Degroeve et al. 2003) se describe un método de selección de atributos usando un algoritmo de estimación de distribuciones.

Los Conjuntos Aproximados y la selección de atributos

Es un hecho reconocido la posibilidad de usar los reductos como selección y reducción de atributos, no solamente en la definición de las relaciones de inseparabilidad que son la base de la teoría de los conjuntos aproximados, sino en otras aplicaciones como el descubrimiento de reglas. Sin embargo, esta beneficiosa alternativa se ve limitada por la complejidad computacional del cálculo de los reductos.

En (Bell and Guan 1998) se muestra que el costo computacional de encontrar un reducto en un sistema de información está acotado por n^2*m^2, donde n es la cantidad de atributos y m es la cantidad de objetos en el universo del sistema de información. Mientras que la complejidad en tiempo de encontrar todos los reductos es $O(2^n*J)$, donde n es la cantidad de atributos, y J es el costo computacional requerido para encontrar un reducto.

La reducción de atributos a través de los conjuntos aproximados se basa en comparar las relaciones de equivalencia o similitud generadas por conjuntos de atributos. Se seleccionan atributos de manera sucesiva hasta que se obtenga un conjunto reducido tal que provea la misma calidad de la clasificación supervisada que el original. Un conjunto de datos puede tener varios conjuntos de reductos. La intersección de todos los conjuntos de reductos se denomina Núcleo (Core) y se refiere al conjunto de atributos indispensables para la

descripción de los objetos (Zhong, Dong et al. 2001). Un objetivo importante en el cálculo de reductos es encontrar un subconjunto mínimo de estos, o sea, un subconjunto reducto con mínima cardinalidad. El problema de encontrar un reducto mínimo en un sistema de información ha sido objeto de varias investigaciones (Alpigini, Peters et al. 2002), algunas basadas en métodos heurísticos (Deogun 1995; Choubey 1996; Bell and Guan 1998; Deogun 1998; Ruiz, Riquelme et al. 2005). La solución más simple para localizar tales reductos sería simplemente generar todos los posibles reductos y escoger aquellos de menor cardinalidad; pero obviamente esta resulta enormemente costosa y poco práctica excepto para conjuntos de datos pequeños.

Se han reportado varios trabajos para encontrar reductos, basados en la Teoría de los Conjuntos Aproximados (Kuo and Yajima 2003; Mi, Wu et al. 2003; Korzes and Jaroszewicz 2005; Caballero and Bello 2006a; Jeon and Jeong 2006). Un método que ha sido muy referenciado es el QuickReduct (Chouchoulas 1999), que trata de calcular un reducto sin generar exhaustivamente todos los posibles subconjuntos de reductos. El algoritmo comienza con un conjunto vacío y adiciona por turno, uno cada vez, a aquellos atributos que resultan tener el mayor grado de dependencia hasta que este produce el máximo valor posible para el conjunto de datos; de esta manera, para una dimensionalidad *n* atributos, se requieren $\left(n^2+n\right)/2$ evaluaciones de la función de dependencia en el peor de los casos. Este proceso, sin embargo, no garantiza encontrar un conjunto mínimo porque al usar la función de dependencia para discriminar candidatos puede conducir la búsqueda hacia un subconjunto que no sea mínimo. Es por eso que sobre esta misma base, otros nuevos enfoques pudieran surgir para tratar de encontrar un buen reducto.

En (Caballero and Bello 2006a) se propone el algoritmo RSReduct para el cálculo de un buen reducto. Este es un "goloso" que comienza por un conjunto vacío de atributos y a través de heurísticas llega a formar un reducto mínimo. Para la construcción de las heurísticas se siguen criterios con respecto a la relevancia (Piñero, Arco et al. 2003), la entropía y la ganancia (Mitchell 1997) de los atributos, así como dependencia entre atributos mediante los Conjuntos Aproximados, y la manipulación de atributos con costos diferentes. Con este algoritmo se obtienen resultados satisfactorios; sin embargo, es un algoritmo costoso. La forma en que se construye la matriz de distinción provoca que la

complejidad de ir chequeando la condición de reducto sea un $O(m^2*n^2)$, para m objetos y n atributos. Por otra parte, el costo computacional de la mejor heurística del algoritmo, es elevado debido a la formación de combinatorias entre atributos.

1.5 Edición y reducción de conjuntos de entrenamiento

Un problema que afecta seriamente la eficacia (en términos de precisión) de los métodos de clasificación supervisada es la presencia de prototipos erróneamente etiquetados en el conjunto de entrenamiento. Durante el entrenamiento pueden producirse errores en el etiquetado o pueden aparecer patrones ruidosos por problemas en la captación de los datos. Estos prototipos suelen aparecer en zonas cercanas a las regiones de decisión e influyen negativamente en el entrenamiento ya que pueden disminuir la precisión de la clasificación supervisada. Además, existe un alto costo computacional asociado a la aplicación de los métodos de clasificación supervisada al conjunto completo de prototipos.

En la figura 1.2 se muestra un esquema funcional de la clasificación supervisada. S_E es un conjunto de prototipos editado, construido a partir de S (conjunto completo de prototipos) mediante algún método de edición. C se refiere al conjunto de referencia (al que se le aplica un método de clasificación supervisada) (Caballero 2005).

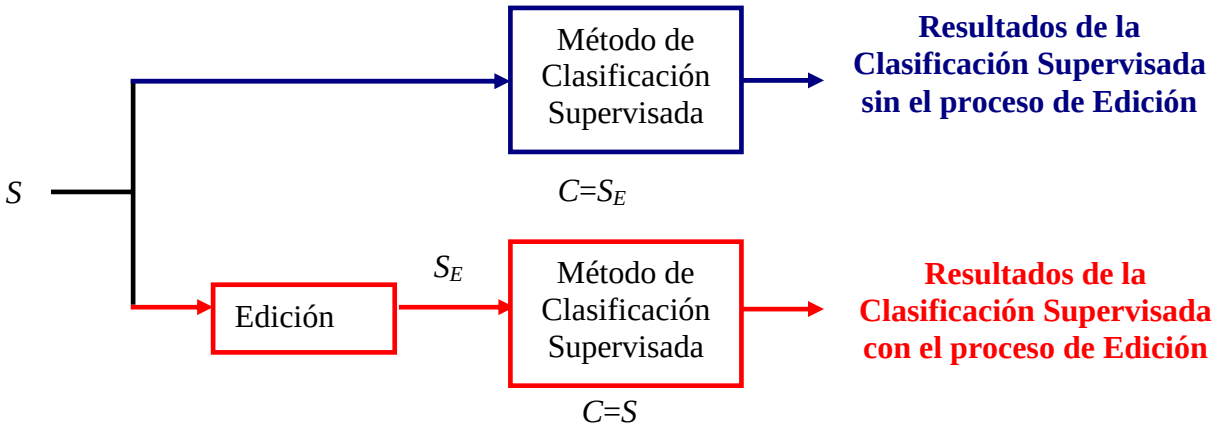

Figura 1.2 Esquema de Clasificación Supervisada

La selección de los objetos de un dominio a incluir en un conjunto de entrenamiento es un problema presente en todos los modelos computacionales que realizan inferencias a partir de ejemplos. El mismo se conoce como edición de conjuntos de entrenamiento. La edición se hace con el objetivo de eliminar los prototipos que inducen a una incorrecta clasificación

supervisada, seleccionando un conjunto de referencia representativo y reducido. Las técnicas de Edición, aunque también producen la eliminación de prototipos, y con ello, la reducción de la matriz de aprendizaje, tienen como objetivo fundamental el obtener una muestra de entrenamiento de mejor calidad para una mejor precisión del sistema. Así, además de conseguir una reducción (en ocasiones notable) del conjunto de referencia, ellas proporcionan un conjunto de "calidad" que hace incrementar la precisión de los métodos de clasificación supervisada cuando se aplican sobre conjuntos editados y decrecer a su vez el costo computacional de dicha clasificación (Devijver and Kittler 1980; Cortijo and Pérez de la Blanca 1997).

Por su parte, las técnicas de Reducción persiguen como objetivo la eliminación de patrones o prototipos para disminuir el tamaño de la matriz de aprendizaje. Se trata de disminuir el trabajo computacional y, a veces, se está dispuesto a pagar con un poco menos de precisión del sistema, pero con más eficiencia computacional. En (Wilson and Martínez 1998; Wilson and Martínez 2000; Ainslie and Sánchez 2002; Lozano, Sánchez et al. 2003; Koggalage and Halgamuge 2004) se tratan indistintamente estos dos términos de edición, reducción y no se tienen en cuenta los aspectos que se analizaron con anterioridad en este documento.

Según (García, Villuendas et al. 2005) los métodos de edición se pudieran agrupar en i) Métodos de selección de objetos (aquellos en los cuales solo se seleccionan objetos del conjunto de entrenamiento siguiendo cierto criterio) y ii) Métodos de construcción de objetos (aquellos en los que los nuevos prototipos pueden no estar en la matriz de entrenamiento).

Métodos de selección de objetos

El método Condensed Nearest Neighbors (**CNN**) es el primer método de selección de objetos reportado en la literatura. Fue desarrollado por Hart en 1966 (Hart 1968). Es un método altamente dependiente del orden de presentación de los objetos. Su propio creador plantea que si el riesgo de Bayes es alto, entonces la lista resultante contendrá esencialmente todos los puntos presentes en el conjunto original y no se llevará a cabo ninguna reducción importante en el tamaño del conjunto (Hart 1968).

El método Reduced Nearest Neighbor (**RNN**), creado por Gates (Gates 1972), trata de eliminar los objetos innecesarios en la lista resultante del método CNN de Hart. Una desventaja de este método es que es dependiente del orden. Según su propio autor (Gates 1972), en muchos casos el tiempo extra de búsqueda de objetos para eliminar no se justifica con el nivel de compactación extra alcanzado.

El Edited Nearest Neighbor (**ENN**) es el primer método de edición reportado en la literatura basado en el error de clasificación. Fue creado por Wilson en 1972 (Wilson 1972) y consiste en la eliminación de todos los objetos que son mal clasificados por el k-NN. Este proceso se realiza por lotes, por lo que los objetos son marcados primero, y después se eliminan simultáneamente todos los marcados.

El método **All-kNN**, desarrollado por Tomek en 1976 (Tomek 1976), es una de las variantes del ENN de Wilson. Consiste en aplicar varias pasadas de k-NN con distintos valores de *k*, y eliminar cualquier objeto que haya sido mal clasificado en cualquiera de ellas. Por la heurística que utiliza, en problemas con clases muy entremezcladas puede eliminar un número excesivo de objetos.

En 1980 surgió el método **Multiedit**, el cual fue desarrollado por Devijver y Kittler (Devijver and Kittler 1980). Consiste en la partición aleatoria de la matriz de entrenamiento en componentes, aplicando después a cada partición un ENN que utiliza un clasificador 1-NN entrenado con la partición siguiente. Después de cada iteración, se unen las particiones resultantes y se repite el proceso hasta que no existan cambios en iteraciones sucesivas.

Según Bezdek (Kuncheva 2004) el método Multiedit trabaja bien en grandes conjuntos de datos. Sin embargo, para conjuntos de datos pequeños, o con clases entremezcladas, puede eliminar todos los objetos de una clase. Si las clases están perfectamente separadas, prácticamente no elimina objetos.

Aha presentó una serie de algoritmos de aprendizaje basado en instancias (Instance-Based Learning, **IBL**) (Aha, Kibler et al. 1991; Aha 1992). Entre estos algoritmos se encuentran los siguientes:

IB1 (Instance Based Llearning Algorithm 1) fue simplemente el algoritmo 1-NN.

IB2, presentado en 1991, es un algoritmo incremental (va adicionando objetos disponibles si cumplen un determinado criterio). Kibler y Aha en 1987, en (Kibler and Aha 1987), llaman a este algoritmo el Algoritmo Creciente o Aditivo (Growth). La lista resultante es construida adicionando a ella los objetos de la matriz de entrenamiento que son mal clasificados con respecto a los elementos que ya están en la lista. IB2 es sumamente sensible al ruido porque los casos ruidosos normalmente serán mal clasificados, y así casi siempre pueden ser retenidos.

En (Ritter 1975; Lowe 1995; Wilson and Martínez 2000; Ainslie and Sánchez 2002; Lozano, Sánchez et al. 2003; Koggalage and Halgamuge 2004) aparecen varias técnicas con el propósito de reducir conjuntos de entrenamiento, basadas en la teoría del Vecino más Cercano. En (Wilson and Martínez 1998) se reportan seis nuevos métodos llamados **DROP 1-5** y **DEL** que pueden ser usados para reducir el número de objetos en conjuntos de entrenamiento. Los algoritmos DROP logran altos índices de compactación. Tienen la desventaja de que la aplicación del ENN y el algoritmo guiado puramente por la no disminución de la eficacia puede obtener conjuntos de objetos que no sean representativos del conjunto original.

Este método tiende a eliminar primero los objetos internos, ya que se ordenan de acuerdo a la distancia a su Vecino más Cercano de clase contraria. La aplicación inicial de un ENN garantiza que los objetos ruidosos sean eliminados. Según su autor (Wilson 1997) este método disminuye las posibilidades de sobreentrenamiento del clasificador.

En los últimos cinco años se han reportado varios métodos de edición. Brighton y Mellish (Brighton and Mellish 2002) introdujeron el método de edición Iterative Case Filtering (**ICF**), el cual es basado en la Regla de los Vecinos más Cercanos y el conjunto de asociaciones de un objeto. En (Olvera, Martínez et al. 2005) se reportan dos esquemas de edición para reducir el tiempo de ejecución del método **BSE**; los autores emplean ENN para filtrar ruidos. El Segundo esquema consiste en re-editar una muestra editada con el propósito de aumentar la precisión de la clasificación supervisada a través de algoritmos DROP.

El método Compact Set Editing (**CSE**) fue enunciado por García y colaboradores en el 2005 (García and Shulcloper 2005). Fue creado para la selección de objetos con datos mezclados e incompletos (MID) y funciones de similitud no necesariamente simétricas. Es un método estable, logrando una compactación de alrededor del 50% para todos los conjuntos de datos, incluso los que tienen clases muy mezcladas. La desventaja principal radica en que no logra niveles de compactación comparables con otros métodos de condensación, incluso en clases totalmente separadas. El método utiliza un nuevo concepto para trabajar, que es de consistencia de subclase, a lo que, en opinión de sus autores, debe el comportamiento estable.

Métodos de construcción de objetos

Koplowitz y Brown en 1978 (Koplowitz and Brown 1978) hicieron una modificación del algoritmo de Wilson, le llamaron **Generalized Editing (GE)**. Ellos estaban preocupados con la posibilidad de eliminar demasiados prototipos del conjunto de entrenamiento en el procedimiento de la Edición de Wilson. Este método consiste en eliminar algunos de los prototipos y además puede cambiar etiquetas de las clases de otros objetos. Se puede considerar como una técnica para modificar la estructura de la muestra de entrenamiento (a través de re-etiquetar algunos de los objetos de entrenamiento) y no solamente para eliminar objetos atípicos (Barandela, Gasca et al. 2002).

Hamamoto y colaboradores en 1997 (Hamamoto, Uchimura et al. 1997) desarrollaron cuatro métodos de **Bootstrap**. Los objetos son construidos por la media de los k-Vecinos más Cercanos de su misma clase, de objetos seleccionados al azar. El usuario define el número de intentos que se realizarán, de los que selecciona el mejor. Según Bezdek (Bezdek and Kuncheva 2004) la familia de Bootstraps permite obtener muy buenos resultados. Este método depende fuertemente del azar y presupone la existencia de un espacio donde se defina una suma entre objetos y el producto de un objeto por un escalar para que las muestras puedan ser promediadas.

El método Learning Vector Quantization with Training Counters (**LVQTC**) creado por Odorico (Odorico 1997) en 1997, es similar al LVQ1 (Kohonen 1995), pero el ajuste de

los objetos va a depender de dos factores: la distancia existente del objeto a un determinado elemento de la matriz de entrenamiento y el historial del objeto.

En (Caballero, Bello et al. 2006b) se propone el algoritmo de edición **Edit2RS** basado en los conceptos de aproximación de la Teoría de los Conjuntos Aproximados, donde se le aplica el algoritmo GE a los elementos del conjunto frontera. Con el método propuesto en (Caballero, Bello et al. 2006b) se obtienen resultados satisfactorios; sin embargo, presenta la deficiencia de que el resultado que arroja el algoritmo Generalized Editing depende del orden de los objetos, pues se van analizando las clases de estos según las de sus k-Vecinos más Cercanos (ver anexo 2).

Diferentes aspectos caracterizan las técnicas de edición y reducción de ejemplares, entre ellos están los siguientes:

Representación: Es necesario decidir si se retiene un subconjunto de los objetos originales o si se modifican ellas usándolas para crear una nueva representación.

Dirección de la búsqueda de las objetos: La construcción del subconjunto S_E a partir del conjunto de entrenamiento S se puede hacer de forma incremental (se comienza con S_E vacío y se van añadiendo casos según algún criterio), decremental (se comienza con S_E=S y se van eliminando ejemplares de S_E según algún criterio) o en lote (se decide si cada objeto cumple el criterio de eliminación antes de separar alguno de ellos).

Tipo de puntos del espacio a retener: El conjunto de casos forma un universo dividido en regiones, se puede tener como criterio retener los objetos situados en las fronteras de las regiones, aquellas situadas en el centro o interior de las mismas, u otro conjunto de puntos.

Volumen de la reducción: Uno de los objetivos de la eliminación de objetos es reducir la memoria de almacenaje necesaria.

Incremento de la velocidad: Otro objetivo es aumentar la velocidad de procesamiento a partir del conjunto de objetos. Típicamente una reducción de la cantidad de objetos producirá una disminución del tiempo de procesamiento.

Precisión de la generalización: El éxito del algoritmo de reducción es poder reducir el número de objetos sin reducir significativamente la capacidad de generalización del algoritmo de procesamiento.

Tolerancia a ruidos: Los algoritmos de reducción también se diferencian respecto a su efectividad en presencia de ruidos en los datos.

Velocidad de aprendizaje: Lo ideal es tener una complejidad de $O(m^2)$ o menos complejo.

Crecimiento incremental: Luego de obtenido el conjunto S_E a partir de S debería ser posible decidir de forma incremental la adición de nuevas instancias al conjunto de entrenamiento a medida que estas se vayan obteniendo.

1.6 Caracterización a priori de los conjuntos de entrenamiento para la clasificación supervisada

Usualmente se evalúa la calidad del conocimiento resultante de aplicar algún método de aprendizaje usando el conjunto de control; es decir, la evaluación es post-aprendizaje. A partir de los datos disponibles se aplican diferentes métodos de aprendizaje para determinar cuál produce un mejor conocimiento. El estudio de la relación entre el conjunto de entrenamiento y la eficiencia y eficacia lograda en el proceso de aprendizaje se realiza usando el método de prueba y error de una forma experimental. Es decir, se realizan sucesivos procesos de entrenamiento y se validan los resultados alcanzados. De modo que resulta de gran interés poder estimar la calidad de los datos antes de proceder al aprendizaje para evitar trabajos innecesarios.

Sin embargo, tanto los métodos de aprendizaje como los métodos para mejorar estos conjuntos dependen de la información original contenida en ellos. El estudio de calidad del conjunto de entrenamiento puede servir de base para tomar decisiones sobre cómo desarrollar el aprendizaje. Es decir, dado un conjunto de entrenamiento (training set, TS) se desea tener una función f tal que $f(TS)$ dé un indicador de cuán bueno parece ser TS para extraer desde él conocimiento necesario para construir un clasificador. El problema es encontrar esa función f.

Existen pocos trabajos en este sentido (Djouadi 1990; Feng and Michie 1994; Michie, Spiegelhalter et al. 1994; Beynon 2003; Caballero, Bello et al. 2004). Uno de ellos es para estimar la calidad, en términos del bias y de la varianza, del conjunto de entrenamiento a través del coeficiente de Bhattacharyya y una distribución gaussiana multivariada. Este trabajo se limita a problemas descritos por dos clases solamente (Djouadi 1990). En 1994, Michie y colaboradores realizan un estudio para estimar, a partir de características de los conjuntos de entrenamiento y medidas estadísticas aplicadas a dichos conjuntos, qué clasificadores supervisados pudieran usarse (Feng and Michie 1994; Michie, Spiegelhalter et al. 1994). Sin embargo, no se obtiene información a priori de la calidad del desempeño de los clasificadores seleccionados. En (Caballero, Bello et al. 2004) se realiza un estudio, a partir de las medidas clásicas de los Conjuntos Aproximados, para estimar la calidad de los conjuntos de entrenamiento para redes Perceptron multicapa.

1.7 El problema de los pronósticos meteorológicos

Un dominio de aplicación donde son abundantes los datos es el relativo a la Meteorología, en el cual se han desarrollado diferentes conjuntos de datos con numerosas variables meteorológicas. Desde comienzos del siglo XX, el problema de la predicción meteorológica se ha abordado de forma numérica, utilizando modelos de circulación atmosférica (sistemas de ecuaciones en derivadas parciales, o aproximaciones de estos) a partir de unas condiciones iniciales conocidas. Los predictores humanos ofrecen diariamente pronósticos regionales y locales, en los núcleos de población, y de fenómenos meteorológicos de interés, principalmente fenómenos producidos en superficie (meteoros como precipitación, temperatura, entre otros).

Por otra parte, se han realizado intentos de predecir meteoros locales aplicando directamente técnicas estadísticas a los registros históricos de observaciones disponibles de estos fenómenos, por ejemplo modelos ARIMA (Autoregresión y de Medias Móviles Integradas). Sin embargo, para determinados elementos meteorológicos estas técnicas tienen grandes limitaciones: i) debido a la resolución de los modelos numéricos que no permiten obtener predicciones locales precisas en puntos geográficos de interés y ii) los modelos estadísticos no incluyen suficiente información sobre los fenómenos físicos que

intervienen en estos procesos. Por ello, se han mostrado ineficientes desde un punto de vista operativo (Cofiño 2004).

Recientemente se han desarrollado aplicaciones automatizadas para la predicción meteorológica (Riordan and Hansen 2002; Cofiño 2004; Sordo 2006). El sistema WIND-1, que se propone en (Riordan and Hansen 2002) en el año 2002, está encaminado a la predicción de la visibilidad e intensidad del viento en terminales aeroportuarias de Nova Scotia, Canadá; los autores implementan un k-NN difuso con el que se combinan elementos del razonamiento basado en casos y la lógica borrosa. En (Cofiño 2004; Sordo 2006) se presentan métodos de predicción local a corto plazo para la zona de Santander, España; se combinan las salidas de los modelos numéricos con la información estadística contenida en las observaciones locales. Estos sistemas son específicos de las zonas locales para las que fueron desarrollados por la diversidad de geografías y características climáticas existentes.

En Cuba se desarrolló el sistema CubaForecast, por el Centro Meteorológico de Cienfuegos en el año 2003 (Díaz and Fernández 2003). Con él es posible realizar, para las diferentes provincias del país, el pronóstico de variables meteorológicas. La información básica de escala sinóptica que utiliza es una maya elaborada por el Centro Europeo de Pronóstico a Mediano Plazo. Este es, según los especialistas meteorólogos, el único sistema automatizado en Cuba con el que se puede realizar el pronóstico de las temperaturas. Con este sistema aunque es cierto que se dio un salto hacia delante, aún no se han obtenido los resultados que se desean.

En el Centro Meteorológico de la provincia de Camagüey se desea pronosticar el rango en que estarán los valores diarios de las temperaturas máximas y mínimas para las seis estaciones meteorológicas de la provincia. Cada una de las estaciones tiene comportamientos diferentes, vinculados con sus especificidades y características geográficas. Se tienen los datos de los valores reales de estas variables y otras asociadas a ella, para los años 2000 al 2006, así como el pronóstico que se realizó en el Centro para cada uno de los días de este período. Considerando la existencia de esos conjuntos de datos, el problema puede ser expresado como un problema de aprendizaje; y como tal ha sido resuelto en esta tesis. En el Capítulo 3 se dan los detalles de la solución a dicho problema.

1.8 Conclusiones parciales

Del análisis crítico de la revisión bibliográfica referente a la selección de atributos relevantes y a la edición de conjuntos de entrenamiento, se concluye que estos métodos no satisfacen a plenitud las necesidades existentes acerca del tratamiento de los datos. Los métodos de selección de atributos son sensibles al uso de una función de evaluación y la calidad de ellos depende en gran medida de la calidad de esa función heurística.

Se ha visto que no hay métodos que sean buenos para todos los problemas, por lo que resulta interesante la combinación de enfoques menos costosos con el uso de funciones heurísticas que no sean dependientes del dominio. La Teoría de los Conjuntos Aproximados es muy simple desde el punto de vista computacional, y este es un aspecto importante para desarrollar métodos que basen su funcionamiento en elementos de esta teoría.

Usualmente el estudio de la relación entre el conjunto de entrenamiento y la calidad lograda en el proceso de aprendizaje se realiza usando el método de prueba y error de una forma experimental; es decir, se realizan sucesivos procesos de entrenamiento y se validan los resultados alcanzados. Este proceso se hace post-aprendizaje, de ahí que resulta muy interesante poder caracterizar a priori los conjuntos de entrenamiento para la clasificación supervisada.

Para el problema de la automatización de la predicción de las temperaturas en el Centro Meteorológico de Camagüey, al cual se le dará solución en el capítulo 3, se hace necesario un preprocesamiento de los datos, tanto para mejorar el costo computacional de las técnicas de clasificación supervisada, como la calidad de las mismas. Con los métodos existentes no se obtienen los resultados deseados, por lo que surge la necesidad de crear nuevos métodos de selección de atributos y edición de conjuntos de entrenamiento, que se proponen en el capítulo 2.

Capítulo 2

La Teoría de los Conjuntos Aproximados para el preprocesamiento de los conjuntos de entrenamiento

2 LA TEORÍA DE LOS CONJUNTOS APROXIMADOS PARA EL PREPROCESAMIENTO DE LOS CONJUNTOS DE ENTRENAMIENTO

Se proponen nuevos métodos para la selección de atributos relevantes y la edición de conjuntos de entrenamiento. En este capítulo se describen estos métodos, así como la propuesta que se construye para la caracterización a priori de los conjuntos de entrenamiento. Se realizan estudios experimentales con conjuntos de datos internacionales.

2.1 Selección de atributos relevantes basada en Algoritmos Evolutivos y la Teoría de los Conjuntos Aproximados

2.1.1 Selección de atributos relevantes usando un enfoque evolutivo

Mientras que en los algoritmos genéticos las interrelaciones entre las variables que representan a los individuos, se tienen en cuenta de manera implícita; en los EDAs dichas interrelaciones se expresan de manera explícita a través de la distribución de probabilidad asociada con los individuos seleccionados en cada generación. De hecho, la estimación de dicha distribución de probabilidad conjunta asociada a los individuos seleccionados en cada generación es la principal dificultad de esta aproximación.

En (Larrañaga and Lozano 2001) se describe la aproximación EDA en tres pasos básicos fundamentales:

- Seleccionar algunos individuos de la población.
- Estimar el modelo computacional subyacente a dichos individuos seleccionados.
- Muestrear la distribución de probabilidad aprendida, con el objetivo de obtener una nueva población de individuos.

Estos pasos son repetidos hasta que se verifique un criterio de parada, previamente establecido.

Diferentes enfoques se han introducido para lograr la estimación de la distribución de probabilidad. Uno de los miembros de esta familia es el Algoritmo de Distribución Marginal Univariado (UMDA, siglas en inglés) para dominio discreto (Mühlenbein 1998), el cual se considera el enfoque más sencillo porque muestrea teniendo en cuenta solamente

probabilidades univariadas. En esta variante, en cada generación, la distribución de probabilidad conjunta, $p_l(x)$, que sirve para estimar el comportamiento de los individuos seleccionados, se factoriza como un producto de distribuciones marginales univariadas e independientes. Es decir:

$$p_l(x) = p(x|D_{l-1}^{Se}) = \prod_{i=1}^{n} p_l(x_i) \tag{2.1}$$

Donde x es un individuo, n es la cantidad de variables (atributos) del individuo, l es el número de la iteración (generación) del proceso, D^{Se}_{l-1} son los individuos seleccionados en la generación *l-1*.

La diferencia principal entre las variantes de UMDA es respecto al dominio de definición (discreto o continuo) de la función objetivo, y a la forma en que el modelo estadístico se estima. En el UMDA discreto la distribución de probabilidad (también discreta) es la que mejor ajusta cada variable de forma independiente en la población seleccionada. Para dominio continuo los parámetros que se estiman son la media y la varianza de cada variable de una distribución normal univariada, esto a partir de la población seleccionada por el algoritmo.

Por otra parte, en el Capítulo 1, se señaló que Wróblewski presentó en (Wróblewski 1995) tres métodos para encontrar reductos cortos, a través de algoritmos genéticos. El primero es un algoritmo genético clásico con individuos representados por cadenas de bits, algunas veces falla para encontrar el óptimo global. El segundo y tercer métodos se basan en algoritmos de permutaciones y el método del "goloso". Su diferencia radica en que el método 2 es un algoritmo "goloso" incremental, mientras que el método 3 es un "goloso" decremental. Los resultados con estos dos métodos son mucho mejores que con el método 1, pero tienen mayor complejidad computacional, según su autor en (Wróblewski 1995).

En el Método 1 la función de adaptabilidad depende de dos parámetros: (L_r) que es el número de *"1"s* en el cromosoma r y (C_r) que es el número de filas cubiertas en la matriz de comparación, ambos valores son normalizados:

$$F(r) = \frac{n - L_r}{n} + \frac{C_r}{k} \qquad (f_1)$$

Donde *n* es la cantidad de atributos y *k* es el número de filas de la matriz de comparación.

En el segundo y tercer métodos la función de adaptabilidad del individuo τ está dada por:

$$F(\tau) = \frac{1}{L_\tau} \qquad (f2, f3)$$

Donde L_τ es la longitud del subconjunto encontrado por el algoritmo "goloso".

En (Wróblewski 1995) se concluye que el método 3 es el que garantiza la obtención de reductos, mientras que con los otros se obtienen conjuntos de atributos que contienen a los reductos, pero no se garantiza que estos conjuntos sean mínimos. Esto es debido a la manera en que cada método obtiene los conjuntos reducidos de atributos. Además, se plantea que solo cuando el conjunto de datos está compuesto por un número pequeño de objetos, el segundo método es superior al tercero.

En esta investigación se propone un método de cálculo de reductos a partir de la integración de las funciones de adaptabilidad (*f1* y *f3*) de los métodos reportados por Wróblewski y el algoritmo UMDA (discreto y continuo).

En la creación de este nuevo método a partir de la combinación de UMDA con *f1*, fue necesario utilizar el UMDA de dominio discreto debido a la codificación que utiliza la función y su representación directa como las variables del problema a resolver por UMDA.

En el caso de la función *f3* que utiliza el algoritmo "goloso" y una permutación para representar las soluciones, se utilizó el algoritmo UMDA para dominio continuo con una pequeña variación en la evaluación de la función objetivo. Para generar las permutaciones, en este caso, cuyas variables son valores reales, se ordenan los valores del vector continuo ascendentemente y se toma como permutación los índices de este ordenamiento. A partir de ahí está lista la permutación para su evaluación en la función.

A continuación se enuncian los pasos fundamentales del algoritmo general.

Algoritmo UMDA+fn

Sea *x* un individuo, *m* cantidad de individuos de la población, *n* la cantidad de atributos, *k* número de iteraciones (generaciones) del proceso, $p_l(x)$ distribución de probabilidad conjunta, y $D^{Se}{}_l$ los individuos seleccionados en la generación *l*.

P1. Inicialización. D_0 ←Seleccionar *m* individuos (la población inicial) al azar.

P2. Repetir estos subpasos en *l=1,2,..,k* iteraciones hasta que se obtenga reducto (criterio de parada). La evaluación se hace teniendo en cuenta una de las funciones definidas anteriormente como *f1* y *f3*.

p2.1. $D^{Se}{}_{l-1}$ ← Seleccionar *t* individuos de D_{l-1}, donde $t \leq m$

p2.2. $p_l(x) = p(x|D_{l-1}{}^{Se}) = \prod_{i=1}^{n} p_l(x_i) = \prod_{i=1}^{n} \frac{\sum_{j=1}^{t} \delta_j(X_i = x_i | D_{l-1}{}^{Se})}{t}$ ←Estimar la distribución de probabilidad conjunta.

p2.3. D_l ← Muestrear *m* individuos (la nueva población) de $p_l(x)$.

Complejidad computacional del algoritmo UMDA+fn

La complejidad más alta de este algoritmo en cada generación es de un $O(m^2*n)$, para *m* individuos de la población y *n* atributos de cada individuo.

- Inicialización: $O(n*m)$
- Evaluación: $O(m*costo(f1 \text{ o } f3))$, donde $costo(f1)$, costo ($f3$): $O(m*n)$
- Selección: $O(m*log(m))$
- Aprendizaje: $O(m*n)$
- Muestreo: $O(m*n)$

Se obtienen resultados satisfactorios, que se muestran en los experimentos 1 y 2 realizados en el epígrafe 2.3.1.

2.1.2 Algoritmos para el cálculo de reductos usando heurísticas y la Teoría de los Conjuntos Aproximados

Los Conjuntos Aproximados se pueden utilizar como una herramienta de selección para descubrir dependencias entre datos y obtener reductos del conjunto inicial de atributos con un mínimo de pérdidas de información.

El Núcleo (Core) de los atributos juega un papel fundamental en la Teoría de los Conjuntos Aproximados para la reducción de atributos (Ye and Chen 2003) y se plantea que constituye uno de los puntos de partida en los procedimientos de reducción de información (Wang 2002). Existen varios trabajos encaminados a poder encontrar este conjunto núcleo (Zhong, Dong et al. 2001; Wang 2002; Ye and Chen 2003), entre otros. Uno de los métodos más referenciados para el cálculo del núcleo es el que utiliza Wang a través del cálculo de la entropía condicional (Wang 2002).

El algoritmo RSRed* surge a partir de estas ideas y encuentra un buen reducto en tiempos aceptables. Parte de un subconjunto de atributos (núcleo que se propone por Wang en el 2002) y construye un reducto a través de la combinación de este subconjunto y atributos que han sido ganadores por determinada heurística. Se proponen dos heurísticas con el objetivo de ir combinando el núcleo con aquellos atributos que sean más relevantes.

La Heurística 1, definida en la expresión 2.2, se basa en los criterios del algoritmo RSReduct (Caballero and Bello 2006a) para la relevancia y ganancia de los atributos.

$$\textbf{\textit{Heurística 1}}\textbf{:} \quad H1(a) = Rl(a) + G(a) \tag{2.2}$$

$Rl(a)$ y $G(a)$ se calculan por las expresiones 2.3 y 2.4 respectivamente:

$$Rl(a) = \sum_{v \in valores(a)} \frac{|S_v|}{|S|} e^{(1-l_v)} \tag{2.3}$$

Donde $0 \leq Rl(a) \leq 1$. $Rl(a)$ es una medida de relevancia del atributo a en el conjunto de objetos S; S_v es el subconjunto de S que contiene los objetos que tienen el valor v en el atributo a; $valores(a)$ es el conjunto de valores diferentes del atributo a; l_v es el número de

clases diferentes presentes en los objetos que tienen el valor *v* para el atributo *a*; $\frac{S_v}{S}$ es la frecuencia relativa de aparición del valor *v* en *S*.

La idea principal de esta medida es maximizar la heterogeneidad entre elementos de clases diferentes y la homogeneidad entre elementos que tienen la misma clase. Consecuentemente, el atributo que maximice *Rl*(*a*) es el mejor o más relevante (Piñero, Arco et al. 2003).

La ganancia radial (*G*(*a*)) propuesta por Quinlan en 1986 (Mitchell 1997) se ha empleado para buscar relaciones entre atributos, como medida alternativa para seleccionar atributos (Caballero and Bello 2006a), para el agrupamiento (Ratke and Andrade 2005), entre otras.

$$G(a) = \frac{Ganancia(S,a)}{Información\ de\ corte(S,a)} \tag{2.4}$$

La *Ganancia* y la *Información de corte* se definen por las expresiones 2.5 y 2.7 respectivamente.

$$Ganancia(S,a) = Entropía(S) - \sum_{v \in valores(a)} \frac{|S_v|}{|S|} Entropía(S_v) \tag{2.5}$$

Donde, *valores*(*a*) es el conjunto de valores posibles para el atributo *a* y S_v es el subconjunto de *S* para el cual *a* tiene el valor *v*. La *Entropía*(*S*) se define como se indica en la expresión 2.6

$$Entropía(S) = \sum_{c=1}^{l} - P_c \log_2 P_c \tag{2.6}$$

Donde, P_c es la proporción de *S* perteneciente a la clase *c* y *l* es la cantidad de clases en *S*.

$$Información\ de\ corte(S,a) = - \sum_{v \in valores(a)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|} \tag{2.7}$$

Donde *valores*(*a*) es el conjunto de los valores del atributo *a*. Esta medida es la entropía de *S* con respecto al atributo *a*.

Sea *Red* un conjunto de atributos candidato a reducto, la Heurística 2, se define en la expresión 2.8. Esta se basa en elementos de la Teoría de los Conjuntos Aproximados, específicamente en el criterio de objetos consistentes respecto a un conjunto de atributos (Zhong, Dong et al. 2001).

***Heurística 2*:** $H2(a) = Consistencia(\mathrm{Re}\, d \cup \{a\}) * Longitud_{\max}(\mathrm{Re}\, d \cup \{a\})$ (2.8)

Donde *Consistencia* y $Longitud_{max}$ se definen en las expresiones 2.9 y 2.10 respectivamente.

$$Consistencia(\mathrm{Re}\, d \cup \{a\}) = \left| POS_{\mathrm{Re}\, d \cup \{a\}}(D) \right| \tag{2.9}$$

Donde $POS_{\mathrm{Re}\, d \cup \{a\}}(D)$ es el conjunto de objetos consistentes de *D* respecto al conjunto de atributos $\mathrm{Re}\, d \cup \{a\}$.

$$Longitud_{\max}(\mathrm{Re}\, d \cup \{a\}) = \max(POS_{\mathrm{Re}\, d \cup \{a\}}(D) \,|\, IND(\{\mathrm{Re}\, d \cup \{a\}, D\})) \tag{2.10}$$

$Longitud_{max}$ representa la longitud máxima de las clases de inseparabilidad (*IND*) (indiscernibilidad) incluidas en la región positiva para el conjunto de atributos $\mathrm{Re}\, d \cup \{a\}$.

Algoritmo RSRed*

Sea un sistema de decisión donde *S* es el conjunto de *m* objetos, *A* el conjunto de atributos que los describe $A=\{a_1,\ldots,a_n\}$, y $D=\{d\}$ atributo de decisión. *Red* subconjunto de atributos de *A* candidato a reducto, *B* subconjunto de atributos de *A* sin seleccionar por el algoritmo y *e*, el umbral de precisión dado por el usuario, para el cual se sugiere un valor de 0.85.

*Si la Heurística que se escoge es **H1**, el algoritmo **RSRed*** es como sigue*:

P1. Inicializar *precisión*=0. Hacer *Red=Núcleo,* donde *Núcleo* representa el subconjunto núcleo de atributos candidatos a reducto del conjunto *A*. Esto se hace a través de los siguientes subpasos (Wang 2002):

p1.1. $Núcleo = \phi$

p1.2. Para cada atributo *a* del conjunto *A*, hacer:

si $H(\{d\}|A) < H(\{d\}|A - \{a\})$ entonces $Núcleo = Núcleo \cup \{a\}$

Donde:

$$H(D|A) = -\sum_{i=1}^{n} p(X_i) * \sum_{j=1}^{m} p(Y_j|X_i) * \log\left(p(Y_j|X_i)\right), \tag{2.11}$$

$$(S|IND(D)) = \{Y_1,...,Y_m\}, \tag{2.12}$$

$$(B \subseteq A) = (S|IND(B)) = \{X_1,...,X_n\}, \tag{2.13}$$

$$p(X_i) = \frac{|X_i|}{|S|},\ p(Y_j|X_i) = \frac{|Y_j \cap X_i|}{|X_i|}, 1 \le i \le n, 1 \le j \le m. \tag{2.14}$$

P2. Hacer $B = A - Red$

P3. Calcular *H1*(*a*) para todos los atributos *a* del conjunto *B*. El atributo que mayor valor obtenga de *H1*(*a*), es el mejor; el objetivo es maximizar *H1*.

P4. Si $precisión \ge e$ entonces parar, donde $precisión = \frac{|POS_{Red}(D)|}{|S|}$.

Si no, si $POS_{\mathrm{Re}d}(D) = POS_A(D)$ entonces *Red* ya cumple la condición de reducto, que significa terminar el algoritmo; de lo contrario, seguir a P5.

P5. Escoger el mejor atributo *a* del conjunto *B*, según la Heurística 1 y hacer:

P5.1. $\mathrm{Re}d = \mathrm{Re}d \cup \{a\}$

P5.2. *B* = *B*-{a}

P6. Regresar al paso P4.

*Si la Heurística que se escoge es **H2**, el algoritmo **RSRed*** es como sigue:*

P1. Inicializar *precisión*=0. Hacer *Red=Núcleo,* donde *Núcleo* representa el subconjunto núcleo de atributos candidatos a reducto del conjunto *A*. Esto se hace a través de los siguientes subpasos (Wang 2002):

p1.1. $Núcleo = \phi$

p1.2. Para cada atributo *a* del conjunto *A*, hacer:

si $H(\{d\}|A) < H(\{d\}|A-\{a\})$ entonces $Núcleo = Núcleo \cup \{a\}$

(ver expresiones 2.11 a 2.14)

P2. Hacer $B = A - Red$

P3. Si $precisión \geq e$ entonces parar, donde $precisión = \frac{|POS_{Red}(D)|}{|S|}$.

Si no, si $POS_{\mathrm{Re}d}(D) = POS_A(D)$ entonces *Red* ya cumple la condición de reducto, que significa terminar el algoritmo; de lo contrario, seguir a P4.

P4. Calcular *H2*(*a*) para todos los atributos *a* del conjunto *B*. El atributo que mayor valor obtenga de *H2*(*a*), es el mejor; el objetivo es maximizar *H2*.

P5. Escoger el mejor atributo que resultó en el paso P4, y hacer:

P5.1. $\mathrm{Re}d = \mathrm{Re}d \cup \{a\}$

P5.2. $B = B$-{a}

P6. Regresar al paso P3.

Complejidad computacional del algoritmo RSRed*

La complejidad más alta de este algoritmo es de un $O(m^2*n)$ para cuando se usa la Heurística 1 y de $O(m^2*n*(n-n_0))$ cuando se usa la Heurística 2, para *m* objetos, *n* atributos y n_0 atributos en el núcleo.

- Cálculo del núcleo: $O(m^2*n)$.
- Cálculo de la Heurística H1: $O(m*n)$.
- Cálculo de la Heurística H2: $O(m*n*(n-n_0))$.

2.2 Algoritmos para la edición de conjuntos de entrenamiento basados en la Teoría de los Conjuntos Aproximados

En la Teoría de los Conjuntos Aproximados resulta de gran interés el significado de la Aproximación Inferior de un sistema de decisión. En la Aproximación Inferior de cada clase de un sistema se agrupan aquellos objetos que con absoluta certeza pertenecen a su

clase. Esto garantiza que los objetos contenidos en la Aproximación Inferior están exentos de ruidos.

La idea básica de aplicar los conjuntos aproximados para editar los conjuntos de entrenamiento es la siguiente: en el conjunto de entrenamiento se colocan los objetos del sistema de decisión inicial que pertenecen a la aproximación inferior de cada clase. Esto es igual a decir que el conjunto de entrenamiento para el primer algoritmo será la región positiva del sistema de decisión. En esta forma, los objetos que están etiquetados incorrectamente o muy cerca de la frontera de decisión pueden ser eliminados del conjunto de entrenamiento, los cuales afectan la calidad de la inferencia. Como en la aproximación inferior de cada clase estarán aquellos objetos que con certeza pertenecen a dicha clase, se garantiza la eliminación de cualquier presencia de ruido en el sistema de decisión.

Algoritmo Edit1RS:

P1. Construir el subconjunto *B*, $B \subseteq A$. Se sugiere que *B* sea un reducto del sistema de decisión para disminuir la dimensionalidad de los atributos sin afectar el proceso de clasificación.

P2. Formar los conjuntos $X_i \subseteq U$, tal que todos los elementos del universo (*U*) que tienen valores d_i en el atributo de decisión están en X_i.

P3. Para cada conjunto X_i, se calcula su aproximación inferior ($R'_*(X_i)$) (descrita en la expresión 1.4) respecto al subconjunto *B* de atributos construido en el paso P1 y la relación *R'*.

P4. Construir el conjunto de entrenamiento editado S_E como la unión de todos los conjuntos $R'_*(X_i)$.

Complejidad computacional del algoritmo Edit1RS

La complejidad más alta del algoritmo Edit1RS es de un $O(m^2*n)$, que es la complejidad de calcular las aproximaciones inferiores para *m* objetos y *n* atributos.

En este método Edit1RS se toman en cuenta sólo los elementos que están en las aproximaciones inferiores. Sin embargo, la información de los objetos que están en la

frontera (BN_B), puede ser interesante, pero habría que cambiar la clase a aquellos que lo requieran, esto pudiera hacerse a través de un proceso de re-etiquetamiento.

Como se dijo en el capítulo 1, el algoritmo de edición Edit2RS (Caballero, Bello et al. 2006b) construye objetos a partir de re-etiquetar aquellos que pertenecen al conjunto frontera. Con este método se obtienen resultados satisfactorios; sin embargo, presenta la deficiencia de que el resultado que arroja el algoritmo Generalized Editing depende del orden de los objetos, pues se van analizando las clases de estos según las de sus k-Vecinos más Cercanos.

La función de pertenencia aproximada (Grabowski 2003), descrita en la expresión 1.10 del capítulo 1, arroja un valor de pertenencia de un objeto a una clase determinada. Si se le aplica esta función a cada uno de los objetos del conjunto frontera para cada una de las clases, se puede saber por cada objeto, a cuál de las clases él tiene mayor grado de pertenencia. Esto constituye una forma de re-etiquetar las clases de los objetos del conjunto frontera. El siguiente algoritmo está propuesto tomando en cuenta estas ideas.

Algoritmo Edit3RS:

P1. Construir el conjunto *B*, $B \subseteq A$. Se sugiere que *B* sea un reducto del sistema de decisión para disminuir la dimensionalidad de los atributos sin afectar el proceso de clasificación.

P2. Formar los conjuntos $X_i \subseteq U$, tal que todos los elementos del universo (*U*) que tienen valores d_i en el atributo de decisión están en X_i.

P3. $S_E = \phi$.

P4. Para cada conjunto X_i:

Calcular su aproximación inferior ($R'_*(X_i)$) (descrita en la expresión 1.4) y su aproximación superior ($R'^*(X_i)$) (descrita en la expresión 1.5), respecto al subconjunto *B* de atributos construido en el paso P1 y la relación *R'*.

p4.1. $S_E = S_E \cup R'_*(X_i)$.

p4.2. $T_i = R'^*(X_i) - R'_*(X_i)$.

P5. Calcular la unión de los conjuntos T_i, para obtener $T = \cup T_i$.

P6. Para cada elemento *x* del conjunto *T*:

p6.1. Calcular la función de pertenencia aproximada a cada clase *X*, según $\mu_X(x) = \frac{|X \cap R'(x)|}{|R'(x)|}$ (ver expresión 1.10).

p6.2. Si la clase con mayor valor (según p6.1) difiere de la clase del objeto, entonces re-etiquetarla por esta clase con la cual se obtuvo mayor valor de pertenencia del objeto.

P7. $S_E = S_E \cup T$. El conjunto de entrenamiento editado se obtiene como el conjunto resultante en S_E.

Complejidad computacional del algoritmo Edit3RS

La complejidad más alta del algoritmo Edit3RS es de un $O(m^2*n)$, para *m* objetos y *n* atributos.

- Cálculo de las aproximaciones inferiores: $O(m^2*n)$
- Cálculo de la región frontera: $O(m^2*n)$
- Cálculo de la medida de pertenencia aproximada para cada objeto de la frontera: $O(m^2*n)$

Caracterización de los algoritmos de edición propuestos

Se puede decir que el método Edit1RS es un método de selección de objetos (selecciona aquellos que están en las aproximaciones inferiores de las clases) y el método Edit3RS es un método además de construcción de objetos, pues no solo selecciona objetos, sino que puede obtener nuevos a partir del proceso de re-etiquetamiento que se aplica a los objetos de la frontera, donde puede ser que algunos de estos cambien su clase.

Los algoritmos Edit1RS y Edit3RS, se pueden caracterizar también de la forma siguiente:

Representación: Retienen un subconjunto de los objetos originales. En el caso del algoritmo Edit3RS, este puede cambiar la clase de algunos objetos.

Dirección de la búsqueda de los objetos: La construcción del subconjunto S_E a partir del conjunto de entrenamiento *S* se realiza en forma de lote. Además, la selección se realiza sobre una visión global del conjunto de entrenamiento, no separada por clases de decisión.

Tipo de puntos del espacio a retener: El algoritmo Edit1RS retiene los objetos situados en el centro o interior de las clases. El algoritmo Edit3RS retiene estos objetos y otros incluidos en las regiones fronteras de las clases.

Volumen de la reducción: El volumen de la reducción depende de la cantidad de inconsistencias en el sistema de información, salvo que el sistema de información sea consistente siempre habrá una reducción del conjunto de entrenamiento.

Incremento de la velocidad: Al disminuir la cantidad de objetos aumenta la velocidad del procesamiento posterior.

Precisión de la generalización: En la mayoría de los casos, uno de los algoritmos o ambos aumentaron significativamente la eficiencia de los clasificadores k-NN, MLP y C4.5.

Tolerancia a ruidos: La Teoría de los Conjuntos Aproximados ofrece un modelo orientado a modelar la incertidumbre dada por inconsistencias, por lo que es efectiva ante la presencia de ruidos. La aproximación inferior elimina los objetos con ruidos.

Velocidad de aprendizaje: La complejidad computacional de hallar la aproximación inferior es $O(n*m^2)$, según (Bell and Guan 1998; Deogun 1998), cercana al valor ideal de $O(m^2)$, y menor que la del cálculo de la cobertura ($O(n^3)$), para *n* cantidad de atributos y *m* cantidad de objetos.

Crecimiento incremental: Son métodos incrementales pues para cada nuevo objeto que aparezca basta determinar si pertenece a alguna aproximación inferior de alguna clase para añadirla o no al conjunto de entrenamiento.

2.3 Estudio experimental para la selección de atributos relevantes y edición de conjuntos de entrenamiento

Para el estudio experimental se utilizó la biblioteca EDALib (Duarte 2003) aplicada al cálculo de reductos con algoritmos evolutivos y se implementaron los softwares RoughSetsReduct 2.0 y EditCE 1.0. El primero incluye los métodos RSReduct (Caballero and Bello 2006a), RSRed*, EBR (Jensen and Qiang 2003) y QuickReduct (Chouchoulas 1999). El segundo software contiene los métodos Edit1RS, Edit3RS y los métodos ENN (Wilson 1972), All-kNN (Tomek 1976), Generalized Editing (GE) (Koplowitz and Brown 1978) y MultiEdit (Devijver and Kittler 1980). Además los métodos propuestos en esta tesis fueron incluidos en la herramienta de Aprendizaje Automatizado Weka[2], como filtros supervisados de atributos y de objetos (instancias), con el objetivo de poderlos comparar con los filtros que esta herramienta implementa y para poder contar con otras facilidades que brinda en el desarrollo del estudio experimental. Se seleccionaron reductos a través del software Rosetta (Ohrn, Komorowski et al. 1998), específicamente con el método SAVGeneticReducer (Vinterbo 1999) y se compararon con otros desarrollados en el presente trabajo.

Para realizar los experimentos de selección de atributos se usaron 23 conjuntos de datos internacionalmente reconocidos, para algunos de los cuales ya existen resultados reportados, obtenidos con otros métodos. Los conjuntos de datos son provenientes del depósito de datos para aprendizaje automatizado disponibles en el sitio ftp de la Universidad de Irvine, California[3] y del sitio personal de Jensen[4], estos son: Balance-Scale, Ballons, Breast-Cancer-Wisconsin, Bupa (Liver Disorders), Credit, Dermatology, E-Coli, Exactly, Hayes-Roth, Heart-Disease (Hungarian), House-Votes, Iris, LED, Lung-Cancer, M-of-N, Monks-1, Mushroom (Agaricus-Lepiota), Pima-Indians-Diabetes, Promoter-Gene-Sequence, Tic-Tac-Toe, Wine Recognition, Yeast y Zoo. Algunos de ellos corresponden a

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

[3] http://www.ics.uci.edu/~mlearn/MLRepository.html

[4] http://www.nd.edu/~rjensen1/research/research.html

datos del mundo real, como por ejemplo House-Votes, Iris, Breast-Cancer-Wisconsin, Pima-Indians-Diabetes, Heart-Disease; otras corresponden a resultados obtenidos en laboratorio como Balloons, Hayes-Roth, LED, M-of-N, Lung-Cancer y Mushroom (Agaricus-Lepiota). Las características de los 23 conjuntos de datos utilizados aparecen en la tabla A1.1 del anexo 1.

De estos 23 conjuntos de datos se seleccionaron los 15 que a continuación se mencionan, para realizar la validación de los experimentos correspondientes a la edición de conjuntos de entrenamiento: Balance-Scale, Breast-Cancer-Wisconsin, Bupa (Liver Disorders), Dermatology, E-Coli, Heart-Disease (Hungarian), Iris, Lung-Cancer, Monks-1, Pima-Indians-Diabetes, Promoter-Gene-Sequence, Tic-Tac-Toe, Wine Recognition, Yeast y Zoo

Se obtuvieron los conjuntos de entrenamiento y control, tomando para el primero el 75% de los casos y para el segundo el 25%, de forma totalmente aleatoria. Siguiendo este principio de selección aleatoria se repitió el proceso diez veces y se obtuvieron para cada conjunto de datos diez conjuntos de entrenamiento y diez conjuntos de control, con el fin de aplicar validación cruzada (cross validation) (Demsar 2006) para una mejor validación de los resultados.

Se definieron varios experimentos, que se describen en los epígrafes 2.3.1 y 2.3.2, para comparar el desempeño de los métodos de selección de atributos y edición de conjuntos de entrenamiento que se proponen en la presente investigación. El estudio experimental se realizó en un procesador Intel de tipo Celerón que funciona a una velocidad de 1.7 GHz, con una memoria RAM de 512 Mb. Esta computadora por sus características de PC comercial no estaba destinada exclusivamente a la corrida de los experimentos.

En los experimentos realizados para la comparación de los algoritmos se utilizaron pruebas no paramétricas, tanto para dos muestras relacionadas (prueba de Wilcoxon) como para *k* muestras relacionadas (prueba de Friedman). Para los casos en que se realizó una prueba de Friedman se aplicó el método de Monte Carlo, con intervalos de confianza del 95% y un número de muestras igual a 10000, mientras que en los casos en los cuales se realizó una prueba de Wilcoxon se aplicó el método de Monte Carlo, con intervalos de confianza del 99% y un número de muestras igual a 10000. Se consideró:

- significativo un resultado de significación menor que 0.05 y mayor que 0.01
- altamente significativa, una significación menor que 0.01
- medianamente significativo, un resultado menor que 0.1 y mayor que 0.05
- no significativo, un resultado mayor que 0.1.

En el procesamiento de los resultados de los experimentos se usó el SPSS versión 13.0.

2.3.1 Resultados experimentales de los algoritmos propuestos para el cálculo de reductos

Parámetros que se usaron en el estudio experimental

Los valores de los parámetros que se utilizaron para los métodos UMDA+f1 y UMDA+f3 fueron: *N = 100*; *g = 3000*; *e = 50*; *T = 0,5*; donde *N* es el número de individuos; *g* es número máximo de evaluaciones que hará; *e* es elitismo, quiere decir que los 50 mejores pasan directo a la próxima generación; *T* es el porciento de mejores que se seleccionan para hacer todos los cálculos.

Los parámetros que se utilizaron para los algoritmos genéticos de Wróblewski fueron: P_c la probabilidad de cruzamiento y P_m la probabilidad de mutación.

Wróblewski (*f1*) ← P_c=0,7 y P_m =0,05

Wróblewski (*f2*, *f3*) ← P_c=0,5 y P_m =0,1

Experimento 1: Comparar los métodos UMDA+f1, UMDA+f3, Wróblewski(*f1*), Wróblewski(*f2*) y Wróblewski(*f3*) (Wróblewski 1995), respecto a las longitudes de los reductos obtenidos y el tiempo que demoran en encontrarlos, para los 23 conjuntos de datos del estudio con sus diez particiones cada uno.

Objetivos: Determinar si los métodos UMDA+f1 y UMDA+f3 son significativamente superiores a los métodos reportados por Wróblewski, respecto a la longitud promedio de los reductos encontrados y al tiempo de ejecución.

Se aplicó una prueba de Friedman entre los métodos UMDA+f1, UMDA+f3 y los reportados por Wróblewski, respecto a la longitud promedio de los reductos encontrados y arrojó que existen diferencias significativas entre ellos. Asimismo, se aplicó una prueba de

Friedman respecto al tiempo de ejecución de estos métodos y también arrojó diferencias significativas entre ellos.

Se aplicó la prueba de Wilcoxon dos a dos, respecto a la longitud promedio de los reductos encontrados y se corrobora lo siguiente: No existen diferencias significativas entre UMDA+f1 y UMDA+f3 en cuanto a la longitud. El promedio de las longitudes de los reductos encontrados por los métodos UMDA+f1 y UMDA+f3 es significativamente inferior al promedio de las longitudes que se obtienen por cada uno de los métodos de Wróblewski.

Se aplicó la prueba de Wilcoxon dos a dos, respecto al tiempo de ejecución para encontrar los reductos y se corrobora lo siguiente: UMDA+f1 es significativamente superior respecto a UMDA+f3. El tiempo que emplean ambos métodos UMDA+f1 y UMDA+f3 es significativamente inferior al que se emplea con los métodos de Wróblewski para obtener los reductos. En el anexo 3 se muestran las tablas de las pruebas estadísticas relacionadas con el experimento 1.

Experimento 2: Comparar las longitudes de los reductos obtenidos por los métodos: UMDA+f1, UMDA+f3, SavGeneticReducer (Ohrn, Komorowski et al. 1998), AttributeSelection (de la herramienta Weka con el evaluador de atributos basado en correlaciones (Hall 1998)), y la longitud del conjunto completo de atributos, para los 23 conjuntos de datos del estudio con sus diez particiones cada uno.

Objetivos: Mostrar que se logran reducciones significativas en el conjunto de atributos cuando se aplican los métodos propuestos de construcción de reductos. Determinar cuáles métodos obtienen menores longitudes en los reductos que construyen.

Se aplicó la prueba de Wilcoxon en las comparaciones entre la longitud de los reductos de cada uno de los métodos propuestos y el conjunto de atributos completo. Aquí se observó, para todos los algoritmos propuestos, que las longitudes de los reductos obtenidos eran menores que el conjunto completo de atributos, con resultados altamente significativos.

Se aplicó una prueba de Friedman entre los métodos UMDA+f1, UMDA+f3, SAVGeneticReducer, AttributeSelection, y no arrojó diferencias significativas entre las longitudes promedio de los reductos que se obtienen a través de estos métodos. Aunque las

diferencias no son significativas, para la mayoría de los conjuntos de datos con los cuales se realizó este experimento, se obtuvieron reductos de menor longitud con los métodos UMDA+f1 y UMDA+f3.

En el anexo 3 se muestran tablas resultantes de las pruebas estadísticas para este experimento 2. La figura A3.1 de este anexo muestra las longitudes de los reductos obtenidos por estos métodos del experimento 2, para algunos de los conjuntos de datos del estudio.

Experimento 3: Comparar los resultados obtenidos por los métodos RSRed*(H1), RSRed*(H2), RSReduct(h_1, h_2, h_3) (Caballero and Bello 2006a), EBR (Jensen and Qiang 2003) y QuickReduct (Chouchoulas 1999), respecto al tiempo empleado para el cálculo de los reductos y la longitud de los reductos encontrados, para los 23 conjuntos de datos y las diez particiones que se realizaron.

Objetivos: Determinar qué heurística (H1, H2) fue la mejor para el método RSRed* respecto al tiempo, así como determinar si existen diferencias significativas entre el tiempo de ejecución de los métodos EBR, QuickReduct, RSReduct y los métodos RSRed*(H1), RSRed*(H2). Determinar qué heurística (H1, H2) fue la mejor para el método RSRed* respecto a la longitud de los reductos encontrados, así como determinar si existen diferencias significativas entre la longitud de los reductos que encuentran los métodos propuestos y el resto de los métodos con los cuales se compara en este experimento 3.

Se aplicó una prueba de Friedman para los métodos RSRed*(H1), RSRed*(H2), RSReduct(h_1, h_2, h_3), EBR y QuickReduct, respecto al tiempo empleado para encontrar el reducto, se demuestra que existen diferencias significativas entre ellos. Se aplicó entonces la prueba de Wilcoxon dos a dos y se obtuvieron los resultados siguientes:

Los métodos RSRed*(H1) y RSRed*(H2) encuentran el reducto en tiempos significativamente menores que el resto de los métodos del estudio y con la Heurística 1 (H1) el RSRed* es significativamente mejor que cuando se aplica la Heurística 2 (H2).

Se aplicó una prueba de Friedman para los métodos RSRed*(H1), RSRed*(H2), RSReduct(h_1, h_2, h_3), EBR y QuickReduct, respecto a las longitudes de los reductos

encontrados, corroborando que también existen diferencias significativas entre ellos. Se aplicó entonces la prueba de Wilcoxon dos a dos y se obtuvieron los resultados siguientes:

El método RSRed*(H2) encontró reductos de longitud más pequeña que el resto de los algoritmos del estudio, de manera significativa. Por su parte, el método RSRed*(H1) obtuvo reductos más cortos que los métodos RSReduct(h_1, h_3) y QuickReduct, con resultados significativos; RSRed*(H1) encontró reductos de longitud similar a los que obtuvo el método RSReduct(h_2), y de longitud mayor que los reductos obtenidos por el método EBR.

En el anexo 3 se muestran algunos resultados arrojados por el experimento 3. La figura A3.2 de este anexo muestra el tiempo empleado para calcular los reductos por los métodos del experimento 3, para algunos de los conjuntos de datos del estudio.

Experimento 4: Comparar la precisión y el error cuadrático medio del clasificador k-NN (IBK del Weka) con todos los atributos de los conjuntos de datos y con los conjuntos de atributos relevantes seleccionados por los métodos UMDA+f1, UMDA+f3, RSRed*(H1), RSRed*(H2), AttributeSelection (de la herramienta Weka con el evaluador de atributos basado en correlaciones (Hall 1998)), SAVGeneticReducer (Ohrn, Komorowski et al. 1998), RSReduct (Caballero and Bello 2006a), EBR (Jensen and Qiang 2003) y QuickReduct (Chouchoulas 1999), para los 23 conjuntos de datos con las diez particiones que se realizaron.

Objetivos: Determinar si existen diferencias significativas en las precisiones y en el error cuadrático medio del clasificador al usar todos los atributos y al tener en cuenta solo un conjunto reducido de estos.

Se aplicó una prueba de Friedman para la precisión del clasificador k-NN con todos los atributos y con algunos de los reductos que se obtuvieron con cada uno de los métodos de selección de atributos. Los resultados arrojaron que no existen diferencias significativas entre las precisiones que se obtuvieron para el clasificador k-NN al realizar este experimento. De igual modo sucedió con el error cuadrático medio. En el anexo 3 se muestran los resultados de la prueba de Friedman para el experimento 4, donde se aprecian estos resultados.

Los algoritmos propuestos disminuyen el costo computacional (menos atributos para el entrenamiento) y sin embargo no disminuyen la precisión del proceso de clasificación supervisada.

Experimento 5: Comparar las longitudes de los reductos obtenidos por los métodos RSRed*(H1), RSRed*(H2), con los resultados que obtuvo Jensen para los métodos EBR y RSAR en (Jensen and Qiang 2003). Comparar las longitudes promedio que arrojan los métodos UMDA+f1, UMDA+f3 con los resultados obtenidos por Jensen con los métodos AntRSAR y GenRSAR (Jensen and Qiang 2003). Estos experimentos se realizan con los mismos 13 conjuntos de datos que analiza Jensen en (Jensen and Qiang 2003).

Objetivos: Determinar cuáles métodos obtienen menores longitudes en los reductos que construyen.

Se aplicó una prueba de Friedman entre los resultados de las longitudes de los reductos arrojadas por los métodos RSRed*(H1), RSRed*(H2), EBR, RSAR y existen diferencias significativas entre ellos. Asimismo, se aplicó la prueba de Friedman para los resultados de los métodos UMDA+f1, UMDA+f3, AntRSAR, GenRSAR y se obtuvieron diferencias significativas entre ellos.

Se aplicó una prueba de Wilcoxon, dos a dos y se detectó que: el método RSRed*(H1) obtiene reductos significativamente más cortos que RSAR; el método EBR obtiene reductos significativamente más cortos que el método RSRed*(H1); el método RSRed*(H2) obtiene reductos significativamente más cortos que RSRed*(H1), EBR y RSAR. Mientras que los métodos UMDA+f1 y UMDA+f3 obtienen reductos más cortos que AntRSAR y que GenRSAR con diferencias significativas. En el anexo 3 se muestran las tablas de las pruebas estadísticas relacionadas con el experimento 5.

Se construye la tabla A3.1 de los anexos a modo de resumen y comparación de los métodos que se proponen en la tesis y otros reportados en la literatura, teniendo en cuenta la longitud de los reductos encontrados y el tiempo empleado en la construcción de estos reductos.

De los métodos UMDA+fn, el UMDA+f1 consume menos tiempo y de manera general, con las dos combinaciones propuestas de UMDA+fn se encuentran reductos de menor longitud que otros métodos basados en técnicas evolutivas. Los UMDA+fn superaron a los

algoritmos genéticos de Wróblewski en la mayoría de los casos, respecto al tiempo de ejecución y la longitud de los reductos encontrados. Por su parte, la Heurística 2 (H2) fue la mejor para el método RSRed* respecto a la longitud del reducto, mientras que la Heurística 1 (H1) obtuvo los reductos en menor tiempo y las longitudes de estos son menores o similares que el resto de los métodos con los cuales se comparó.

2.3.2 Resultados experimentales de los algoritmos propuestos para la edición de conjuntos de entrenamiento

Parámetros que se usaron en el estudio experimental

- Para los métodos ENN, Todo k-NN, Generalized Editing, Multiedit y Edit3RS, los cuales aplican en su totalidad la regla k-NN en el momento de la edición, se tomó el valor de k=10.
- Para el método Multiedit se seleccionó N_p=3 (número de particiones); N_i=5 (número de iteraciones).
- Para los métodos Edit1RS y Edit3RS se usó como umbral de similitud un valor mayor o igual a 0.7 para todas las bases del estudio. La función de similitud usada para cada uno de los atributos estuvo en correspondencia con el tipo del atributo (Wilson and Martínez 1997; García 2003).

Experimento 6: Comparar los resultados de precisión general y el error cuadrático medio del conjunto de control, obtenidos por los clasificadores k-NN (IBK del Weka) y la red neuronal Perceptron multicapa (Multilayerperceptron del Weka) para muestras de entrenamiento sin editar, editadas por Edit1RS, Edit1RS (con reducto), Edit3RS, Edit3RS (con reducto) y editadas por los métodos clásicos de edición: ENN (Wilson 1972), All-kNN (Tomek 1976), Generalized Editing (Koplowitz and Brown 1978) y Multiedit (Devijver and Kittler 1980), para los 15 conjuntos de datos y las diez particiones que se realizaron.

Objetivos: Mostrar que la precisión general del conjunto de control es mayor cuando se usan muestras de entrenamiento editadas por los métodos de edición que se proponen, que cuando se usan muestras de entrenamiento sin editar; y que además, el error cuadrático medio es menor al usar los conjuntos editados obtenidos a través de los métodos propuestos

en esta investigación. Determinar si existen diferencias en las precisiones y en el error cuadrático medio cuando se edita por Edit1RS, Edit1RS (con reducto), Edit3RS, Edit3RS (con reducto). Determinar las diferencias que existen entre los resultados de las precisiones y del error cuadrático medio, obtenidos con muestras editadas por los métodos propuestos y editadas por los métodos clásicos.

Se aplicó la prueba de Wilcoxon para los resultados de la precisión general del conjunto de control con el clasificador k-NN, para muestras de entrenamiento sin editar y las editadas por cada uno de los métodos y los resultados arrojaron que las precisiones son significativamente superiores cuando se editan las muestras que cuando se usan muestras sin editar. Se realizó un análisis estadístico similar con el error cuadrático medio y se obtuvo que el error cuadrático medio es significativamente menor con muestras editadas que sin editar. Similar a lo que sucedió con el k-NN, sucedió con el clasificador supervisado Perceptron multicapa. Ver tablas del experimento 6 en el anexo 4.

Se aplicó la prueba de Wilcoxon para las precisiones con las muestras editadas por Edit1RS y Edit1RS (con reducto) y no se aprecian diferencias significativas, de igual modo sucedió con Edit3RS y Edit3RS (con reducto), para los dos clasificadores supervisados del experimento.

Se aplicó una prueba de Friedman para las precisiones obtenidas con el conjunto de control para el clasificador k-NN, con muestras de entrenamiento editadas por el Edit1RS, Edit3RS, ENN, All-kNN, Generalized Editing y Multiedit y arrojó que existen diferencias significativas entre ellas. Se aplico una prueba de Wilcoxon, dos a dos, y se verificó lo siguiente: Los métodos ENN, All-kNN y Generalized Editing obtienen resultados significativamente inferiores que los métodos Edit1RS y Edit3RS, mientras que los resultados de Multiedit no son significativamente diferentes que los de Edit1RS y Edit3RS. El método Edit3RS obtiene mejores resultados que Edit1RS. Lo mismo sucedió con el clasificador MLP. Sin embargo, al realizar el análisis estadístico para el error cuadrático medio, tanto del clasificador k-NN como del Perceptron multicapa, se obtienen resultados significativamente inferiores con muestras editadas por Edit1RS y Edit3RS que con todas aquellas editadas por los métodos clásicos, y además, se obtuvieron resultados significativamente inferiores para el error cuadrático medio con el método Edit3RS que con

el Edit1RS. Ver tablas del experimento 6 del anexo 4. La figura A4.1 de este anexo muestra las precisiones obtenidas por el calsificador k-NN para algunos de los conjuntos de datos del estudio, al ser editados por los métodos del experimento 6.

Experimento 7: Comparar las muestras editadas por los métodos Edit1RS, Edit1RS (con reducto), Edit3RS, Edit3RS (con reducto), ENN (Wilson 1972), All-kNN (Tomek 1976), Generalized Editing (Koplowitz and Brown 1978) y Multiedit (Devijver and Kittler 1980), respecto al porciento de reducciones de objetos respecto al conjunto original, para los 15 conjuntos de datos y las diez particiones que se realizaron.

Objetivos: Determinar si existen diferencias significativas respecto al porciento de reducciones de los objetos de los métodos Edit1RS, Edit1RS (con reducto), Edit3RS, Edit3RS (con reducto), ENN, All-kNN, Generalized Editing y Multiedit.

Se aplicó la prueba de Friedman para las reducciones de los métodos Edit1RS, Edit1RS (con reducto), Edit3RS y Edit3RS (con reducto), el cual arrojó que existen diferencias significativas entre ellos. Se aplicó la prueba de Wilcoxon, dos a dos, la cual arrojó lo siguiente: Edit1RS logra porcientos de reducción significativamente mayores que Edit3RS, no existen diferencias significativas entre Edit1RS y Edit1RS (con reducto), ni entre Edit3RS y Edit3RS (con reducto).

Se aplicó la prueba de Wilcoxon, dos a dos, para las reducciones que se obtienen por los métodos Edit1RS, Edit3RS, ENN, All-kNN, Generalized Editing y Multiedit y se corrobora que: Edit1RS obtienen porcientos de reducción significativamente mayores que los métodos ENN, All-kNN, Generalized Editing. Ver tabla del anexo 4 referida al experimento 7. En la figura A4.2 de este anexo se observan las reducciones de muestras editadas por los métodos del experimento 7 para algunos de los conjuntos de datos del estudio.

Experimento 8: Comparar los métodos Edit1RS, Edit1RS (con reducto), Edit3RS, Edit3RS (con reducto), ENN (Wilson 1972), All-kNN (Tomek 1976), Generalized Editing (Koplowitz and Brown 1978) y Multiedit (Devijver and Kittler 1980), respecto al tiempo de ejecución de los mismos, para los 15 conjuntos de datos y las diez particiones que se realizaron.

Objetivos: Determinar si existen diferencias significativas entre los métodos Edit1RS, Edit1RS (con reducto), Edit3RS, Edit3RS (con reducto), ENN, All-kNN, Generalized Editing y Multiedit, respecto al tiempo de ejecución.

Se aplicó la prueba de Friedman para el tiempo de ejecución de los métodos Edit1RS, Edit1RS (con reducto), Edit3RS y Edit3RS (con reducto), ENN, All-kNN, Generalized Editing y Multiedit, el cual arrojó que existen diferencias significativas entre ellos. Se aplicó la prueba de Wilcoxon, dos a dos, y se corrobora lo siguiente: el método Edit1RS obtiene los resultados en menor tiempo que los métodos Edit3RS, All-kNN, Generalized Editing y Multiedit. Por su parte, el método Edit3RS obtiene las muestras editadas en menores tiempos que el método All-kNN. En el anexo 4 se observan los resultados de este experimento. La figura A4.3 de este anexo muestra el tiempo requerido para editar por los métodos del experimento 8 para algunos conjuntos de datos del estudio.

Experimento 9: Comparar los resultados de precisión general y precisión por clases del conjunto de control, así como los valores del error cuadrático medio que se obtienen con el clasificador k-NN (IBK del Weka) para muestras de entrenamiento editadas por Edit1RS, Edit3RS y editadas por el filtro de selección de instancias del Weka (Resample), para los 15 conjuntos de datos y las diez particiones que se realizaron.

Objetivos: Determinar si existen diferencias tanto en la precisión general, como por clases, y en el error cuadrático medio del clasificador k-NN cuando se usan muestras de entrenamiento editadas por los métodos de edición que se proponen, que cuando se usan muestras de entrenamiento editadas por el Resample.

Se aplicó la prueba de Friedman para las precisiones obtenidas por el conjunto de control con el clasificador k-NN para muestras de entrenamiento editadas por Edit1RS, Edit3RS y Resample, y arrojó que existen diferencias significativas entre ellos. Se aplicó la prueba de Wilcoxon, dos a dos, y se corroboró lo siguiente: Con las muestras de entrenamiento editadas por Resample se obtienen, tanto para la precisión general de la clasificación supervisada como por clases, resultados significativamente superiores que con el método Edit1RS, mientras que se obtienen resultados significativamente inferiores del método Resample respecto al método Edit3RS. Mientras que al realizar el análisis estadístico

respecto al error cuadrático medio se obtuvieron los siguientes resultados: Con muestras editadas por los métodos Edit1RS y Edit3RS se obtienen resultados significativamente menores del error cuadrático medio para el clasificador k-NN, que los obtenidos con las muestras editadas por el método Resample. Ver tablas del experimento 9 en el anexo 4.

Experimento 10: Comparar los resultados de precisión general y precisión por clases del conjunto de control, así como del error cuadrático medio, obtenidos por el clasificador k-NN para muestras de entrenamiento editadas por Edit1RS, Edit3RS y editadas por los métodos de edición IB1 e IB2 de Aha (Aha, Kibler et al. 1991), para los 15 conjuntos de datos y las diez particiones que se realizaron.

Objetivos: Determinar si existen diferencias tanto en la precisión general, como por clases, y en el error cuadrático medio del clasificador k-NN cuando se usan muestras de entrenamiento editadas por los métodos de edición que se proponen, que cuando se usan muestras de entrenamiento editadas por los métodos de edición IB1 e IB2.

Se aplicó la prueba de Wilcoxon, dos a dos, para las precisiones del conjunto de control, tanto general como por clases, del clasificador k-NN para las muestras de entrenamiento editadas por los métodos Edit1RS, Edit3RS, IB1 e IB2, y se obtienen los siguientes resultados: para el análisis de la precisión general con el método IB1 se logran resultados significativamente inferiores que con los métodos Edit1RS y Edit3RS; mientras que respecto a la precisión por clases no se detectan diferencias significativas entre los métodos IB1 y Edit1RS, pero IB1 obtiene resultados significativamente inferiores que Edit3RS y no existen diferencias significativas entre los métodos propuestos y el IB2. Sin embargo, respecto al error cuadrático medio del clasificador k-NN, los métodos Edit1RS y Edit3RS son superiores, pues con las muestras editadas por ellos, se obtienen resultados del error cuadrático medio significativamente inferiores que los que se obtienen con las muestras que fueron editadas por IB1 e IB2. Ver tabla del experimento 10 en el anexo 4.

Experimento 11: Comparar los métodos Edit1RS y Edit3RS con los métodos que se reportan por Olvera en el 2005 (Olvera, Martínez et al. 2005): BSE, ICF, DROP3, DROP4, ENN+BSE, DROP3+BSE, DROP4+BSE, DROP5+BSE, con respecto a la precisión general de la muestra de control del clasificador k-NN cuando se emplean muestras de

entrenamiento editadas por estos métodos y respecto a las reducciones de los objetos respecto a la muestra original. Estos experimentos se realizan para los mismos diez conjuntos de datos que analiza Olvera.

Objetivos: Determinar si existen diferencias significativas entre los métodos Edit1RS, Edit3RS y los que se reportan en (Olvera, Martínez et al. 2005), respecto a la precisión del clasificador k-NN y las reducciones de la muestra.

Se aplicó la prueba de Friedman para las precisiones de los métodos mencionados en el Experimento 10 y arrojó que existen diferencias significativas entre ellos. Se aplicó la prueba de Wilcoxon, dos a dos, para dichas precisiones y se obtuvieron los resultados siguientes: con las muestras editadas por los métodos BSE y DROP4 se obtiene resultados de precisión del clasificador k-NN significativamente mayores que con el método Edit1RS, mientras que con el Edit3RS se obtuvieron resultados significativamente superiores a los obtenidos por DROP4. Los resultados con el método Edit1RS fueron significativamente superiores que con los métodos ICF y DROP3, mientras que los resultados con el método Edit3RS fueron significativamente superiores que con los métodos ENN+BSE, DROP3+BSE, DROP4+BSE, DROP5+BSE, ICF y DROP3. Ver tablas del experimento 11 en el anexo 4.

Se aplicó el test de Friedman para los porcientos de reducción de los métodos Edit1RS, Edit3RS y los mencionados en (Olvera, Martínez et al. 2005) y arrojó diferencias significativas entre ellos. Se aplicó el test de Wilcoxon, dos a dos, para estos métodos y los resultados fueron: no existen diferencias significativas entre los métodos ICF y Edit1RS, mientras que con los métodos BSE, DROP3, DROP4, ENN+BSE, DROP3+BSE, DROP4+BSE, DROP5+BSE se obtienen porcientos de reducciones significativamente mayores que con los métodos Edit1RS y Edit3RS.

La tabla A4.1 de los anexos muestra el comportamiento de los métodos de edición propuestos, teniendo en cuenta la precisión de la clasificación, reducciones de la muestra original y tiempo de ejecución, con respecto a otros métodos reportados en la literatura.

Los dos métodos propuestos obtienen resultados satisfactorios respecto a la precisión de los clasificadores, con las muestras editadas por el Edit3RS se obtienen mayores precisiones

que con las editadas por Edit1RS. Sin embargo, el Edit1RS reduce más la muestra de entrenamiento que el Edit3RS. Ambos métodos son muy simples computacionalmente por lo que el tiempo requerido para editar las muestras de entrenamiento es menor o igual que el que emplea la mayoría de los métodos con los cuales se comparó.

2.4 Caracterización a priori de los conjuntos de entrenamiento basada en las medidas de estimación de los Conjuntos Aproximados

En este epígrafe se proponen nuevas medidas de la Teoría de los Conjuntos Aproximados y se desarrolla una propuesta para, a partir de estas, predecir a priori la calidad de un conjunto de entrenamiento, y además poder seleccionar qué tipo de clasificador supervisado será el más conveniente usar en el proceso de aprendizaje: una red neuronal (Perceptron multicapa), un árbol de decisión (C4.5) o un método de aprendizaje perezoso (k-NN).

2.4.1 Nuevas medidas para evaluar los sistemas de decisión, basadas en la Teoría de los Conjuntos Aproximados

En esta tesis se proponen variantes generalizadas tanto para la precisión como para la calidad de la clasificación supervisada, porque en muchas aplicaciones los expertos pueden ponderar las clases o se pueden seguir heurísticas para definir en qué medida las clases son importantes.

Precisión generalizada de la clasificación. En esta medida por cada clase se da la posibilidad de considerar un peso que influya en la evaluación del sistema de decisión.

$$\mathrm{A}_G(DS) = \frac{\sum_{i=1}^{l} \left(\alpha(X_i) \cdot w(X_i)\right)}{\sum_{i=1}^{l} w(X_i)} \tag{2.15}$$

Donde *l* es la cantidad de clases del sistema de decisión, $\alpha(X_i)$ se calcula como se indica en la expresión 1.7. En la expresión 1.10 se definió la función de pertenencia aproximada; el cálculo de la media de esta función aplicada a los objetos de la clase, puede ser considerada como una medida de importancia de la clase, es decir, el valor para $w(X_i)$.

Calidad generalizada de la clasificación. En esta expresión también se permite ponderar por clases al promediar la calidad de cada aproximación.

$$\Gamma_G(DS) = \frac{\sum_{i=1}^{l} \left(\gamma(X_i) \cdot w(X_i)\right)}{\sum_{i=1}^{l} w(X_i)} \tag{2.16}$$

Donde *l* es la cantidad de clases del sistema de decisión, $\gamma(X_i)$ se calcula como se indica en la expresión 1.8. En (Caballero, Arco et al. 2007) se propone la función de compromiso aproximado, esta cuantifica en qué grado la *R'*(*x*) (clase de similitud de *x*) cubre la clase *X*, y está dada por la expresión 2.17.

$$\nu_X(x) = \frac{\left|X \cap R'(x)\right|}{\left|X\right|} \tag{2.17}$$

La media del compromiso aproximado de los objetos a la clase, puede ser considerada como un valor para ser asignado al peso de cada clase ($w(X_i)$), en la expresión 2.16.

En las expresiones (2.15) y (2.16), $w(X_i)$ es el peso de la clase X_i y es un valor entre 0 y 1, este valor puede ser definido por los expertos. El hecho de considerar la media de la pertenencia aproximada y el compromiso aproximado por clases, respectivamente, constituye una manera de dar valor a estos $w(X_i)$.

Resulta interesante no solo considerar los resultados experimentales de medidas que describen a los datos por clases, sino también tener una manera de cuantificar el comportamiento del conjunto de datos de manera general. Por tal motivo, se propone el coeficiente de aproximación general, definido por la expresión 2.18. Este coeficiente muestra una proporción entre la cantidad de objetos que pudieran ser bien clasificados (aquellos que pertenecen a la aproximación inferior de las clases) y la cantidad de objetos que pudieran o no pertenecer a las clases del sistema de decisión.

Coeficiente de aproximación general. Se evalúa la calidad del sistema de información sin diferenciar por clases.

$$T(DS) = \frac{\sum_{i=1}^{l} \left| R'_{*}(X_i) \right|}{\sum_{i=1}^{l} \left| R'^{*}(X_i) \right|} \qquad (2.18)$$

Donde l es la cantidad de clases del sistema de decisión, $R'_{*}(X_i)$ es la aproximación inferior (descrita en la expresión 1.4) y $R'^{*}(X_i)$ su aproximación superior (descrita en la expresión 1.5).

2.4.2 Estudio experimental de la relación entre las medidas de inferencia y el desempeño de los clasificadores

Para el estudio experimental se seleccionaron los conjuntos de datos siguientes: Balance-Scale, Breast-Cancer-Wisconsin, Bupa (Liver Disorders), Dermatology, E-Coli, Heart-Disease (Hungarian), Iris, Lung-Cancer, Monks-1, Pima-Indians-Diabetes, Promoter-Gene-Sequence, Tic-Tac-Toe, Wine Recognition, Yeast, Zoo, Glass, Hayes-Roth, Soybean, Ionosphere, Page-blocks, Postoperative, Waveform, Credit-Screening, Hepatitis y Lymphography. La descripción de estos conjuntos de datos aparece en la tabla A1.1 del anexo 1.

El esquema siguiente de estudio experimental se aplicó a cada conjunto de datos estudiado:

1. Formación de muestras de entrenamiento y control.

Se obtuvieron los conjuntos de entrenamiento y control, siguiendo el principio de validación cruzada, se selecciona el 75% de los objetos para el entrenamiento y el 25%, para conjunto control.

2. Proceso de clasificación supervisada.

Se realizó el proceso de clasificación supervisada a través de la herramienta Weka, específicamente con los clasificadores supervisados k-Vecinos más Cercanos (IBK), red neuronal Perceptron multicapa y C4.5 (J48).

3. Cálculo de las Medidas de Inferencia.

Para cada conjunto de entrenamiento se calcularon las medidas: Precisión de la aproximación (expresión 1.7), Calidad de la aproximación (expresión 1.8), Calidad de la

clasificación (expresión 1.9), Precisión generalizada de la clasificación (expresión 2.15), Calidad generalizada de la clasificación (expresión 2.16) y Coeficiente de aproximación general (expresión 2.18), descritas en los epígrafes 1.2 y 2.4. El peso ($w(X_i)$) que se utilizó para calcular las medidas definidas en las expresiones 2.15 y 2.16 fueron la media de la pertenencia aproximada por clases y la media del compromiso aproximado por clases, respectivamente. Se implementó el sistema MIRST1.0 que permite el cálculo de estas medidas de inferencia.

4. Cálculo de correlaciones estadísticas.

Con ayuda del SPSS 13.0 se buscan los coeficientes de Correlación de Pearson entre las medidas de inferencia descritas en las expresiones 1.7, 1.8 y el desempeño por clases de los clasificadores; así como los coeficientes de correlación entre las medidas descritas en las expresiones 1.9, 2.15, 2.16, 2.18 y los resultados del desempeño general de la clasificación supervisada. Se llega a coeficientes de correlación en todos los casos cercanos a 1, con una significación bilateral menor que 0.01, con lo que se puede asegurar que el coeficiente de correlación es significativo ($p<0.01$). Las correlaciones estadísticas obtenidas a través del SPSS 13.0 se pueden apreciar en las tablas A5.1 a la A5.3 y tablas A6.1 a la A6.3 de los anexos 5 y 6.

2.4.3 Generación del conocimiento a partir de las medidas de inferencia usando métodos de aprendizaje automatizado

Resulta interesante poder inferir conocimiento a partir de estos resultados, es decir, que para un determinado conjunto de entrenamiento, sea posible decidir cuál clasificador supervisado (entre k-NN, MLP y C4.5) es más conveniente aplicar y poder además, dar una valoración cualitativa del resultado de la precisión general que arrojará dicho clasificador para ese conjunto de entrenamiento (baja, media, alta, muy alta). Para esto se ha utilizado el generador de reglas C4.5 y además una red Perceptron multicapa, y se comparan los resultados arrojados por ambos. Además, una vez que se haya seleccionado el clasificador supervisado idóneo para un conjunto de entrenamiento determinado, se puede inferir el valor de su precisión, a través del método estimador de funciones k-Vecinos más Cercanos (k-NN).

Construcción de conjuntos de datos para el entrenamiento

Para esto se crearon seis nuevos conjuntos de datos, dos correspondientes a cada clasificador supervisado estudiado: un conjunto de datos con los valores de la precisión discretizados (figura A7.1 anexo 7) y otro con los valores numéricos de la precisión (figura A7.2 anexo 7).

Estos conjuntos de datos tendrán cuatro atributos predictores, representados por las medidas: Calidad de la clasificación, Precisión generalizada de la clasificación, Calidad generalizada de la clasificación y Coeficiente de Aproximación General, todos con valores entre cero y uno. El atributo objetivo (clase), para los tres conjuntos de datos con la precisión discretizada, tendrá los valores: A→ "No aplicable, precisión muy baja"; B→ "Aplicable, precisión baja"; C→ "Aplicable, precisión media"; D→ "Aplicable, precisión alta" y E → "Aplicable, precisión muy alta".

Para poder categorizar la precisión de los clasificadores del estudio en A, B, C, D y E, se calcularon los percentiles de la precisión de cada clasificador a través del SPSS 13.0 y se obtuvieron los resultados que se muestran en la tabla A7.1 del anexo 7. De esta manera se asociaron los valores de la precisión a las diferentes categorías según: A← valores inferiores al 20 percentil; B← valores entre el 20 y 40 percentil; C← valores entre el 40 y 60 percentil; D← valores entre el 60 y 80 percentil y E← valores mayores que el 80 percentil.

Aquí se pueden aplicar otras técnicas de discretización para categorizar la precisión del clasificador en A, B, C, D, E, por ejemplo el método de dicretización basado en la entropía, discretización basada en el error (Witten and Frank 2005b).

En las figuras A7.1 y A7.2 del anexo 7 se muestran los conjuntos de datos que se construyeron a partir de los resultados de las medidas de inferencia y la precisión de los clasificadores. Cada conjunto de datos está formado por 250 ejemplos, pues a cada uno de los 25 conjuntos de datos con los cuales se realizó el estudio, se le dividió aleatoriamente en diez muestras de entrenamiento y prueba.

Generación de reglas de forma automatizada

A través del método C4.5 se realiza el proceso de clasificación supervisada, con el objetivo de determinar, para un nuevo conjunto de datos (nuevo ejemplo), una valoración cualitativa de la precisión que obtendrían los clasificadores que se estudiaron. Aquí se utilizan para el entrenamiento los tres conjuntos de datos que se muestran en la figura A7.1 del anexo7.

Los resultados del desempeño del C4.5 para estimar la precisión de los clasificadores a partir de las medidas propuestas, se muestran en la tabla 2.1. Estos son los resultados de aplicar una validación cruzada para diez muestras de prueba.

Tabla 2.1 Medidas de desempeño general al usar el método C4.5

Medidas de desempeño general	Para precisión de los clasificadores		
	k-NN	**MLP**	**C4.5**
Instancias clasificadas correctamente (%)	95,906	95,313	96,491
Estadígrafo de Kappa	0,909	0,897	0,923
Media del error absoluto	0,045	0,050	0,050
Raíz del error cuadrático medio	0,164	0,203	0,185
Error absoluto relativo (%)	9,811	10,976	10,943

Clasificación supervisada usando una red neuronal Perceptron multicapa

Se construye una red neuronal con cuatro neuronas en la capa de entrada, una capa oculta y cinco neuronas en la capa de salida, una para cada categoría. De esta manera se tiene una red neuronal que se entrena con tres conjuntos de datos (uno para cada clasificador del estudio con el atributo objetivo discretizado (clase)), capaz de decidir dado un nuevo conjunto de entrenamiento si es aplicable o no a cada clasificador, y en caso de ser aplicable, además da una valoración de la precisión que se obtendrá (baja, media, alta, muy alta). El desempeño de esta red neuronal para la prueba de validación cruzada (diez particiones) se muestra en la tabla 2.2

Tabla 2.2 Medidas de desempeño general al usar el Perceptron multicapa

Medidas de desempeño general	Para precisión de los clasificadores		
	k-NN	**MLP**	**C4.5**
Instancias clasificadas correctamente (%)	92,949	92,308	91,667
Estadígrafo de Kappa	0,876	0,863	0,857
Media del error absoluto	0,070	0,081	0,081
Raíz del error cuadrático medio	0,184	0,219	0,199
Error absoluto relativo (%)	18,58	21,355	21,303

Cuando se presenta un nuevo conjunto de entrenamiento, es decir un nuevo ejemplo, para predecir a priori el clasificador idóneo a aplicar, se procede según los pasos siguientes:

P1. Se particiona aleatoriamente el nuevo ejemplo en conjunto de entrenamiento (75%) y prueba (25%), esto se realiza diez veces y de esta forma se obtienen diez conjuntos de entrenamiento y diez de prueba.

P2. A cada uno de los diez conjuntos de entrenamiento, obtenidos en el paso P1 se le calculan las medidas: Calidad de la clasificación, Precisión generalizada de la clasificación, Calidad generalizada de la clasificación y Coeficiente de Aproximación General. Se obtiene para cada medida el valor promedio de los diez resultados.

P3. El nuevo ejemplo ahora está descrito por cuatro atributos asociados a las medidas, cuyos valores son los resultados de las medias obtenidas en el paso P2.

P4. Se determina para cada clasificador (k-NN, MLP, C4.5) la calidad de la precisión según las clases (A→ "No aplicable, precisión muy baja"; B→ "Aplicable, precisión baja"; C→ "Aplicable, precisión media"; D→ "Aplicable, precisión alta" y E → "Aplicable, precisión muy alta"). Esto puede realizarse a través del C4.5 o el Perceptron multicapa, ambos entrenados con los conjuntos de datos que se muestran en la figura A7.1 del anexo 7.

P5. Se escoge como clasificador idóneo aquel que según el paso P4 le corresponda mejor valor de la clase, en el orden (E, D, C, B, A).

Estimación del valor de la precisión del clasificador más apropiado

Luego de elegir el clasificador con el que se obtendrá más alta precisión para un conjunto de entrenamiento dado, se estima este valor a partir de los conjuntos de datos de la figura A7.2 del anexo 7, a través del método k-NN. Se calcula el valor de *k* óptimo y se obtiene un valor aproximado de la precisión del clasificador seleccionado. Los resultados obtenidos después de realizar una validación cruzada (diez muestras de prueba) se pueden observar en la tabla 2.3.

Tabla 2.3 Medidas de desempeño general al usar el método k-NN

Medidas de desempeño general	Para precisión de los clasificadores		
	k-NN	**MLP**	**C4.5**
Coeficiente de correlación	0,969	0,914	0,913
Media del error absoluto	0,004	0,040	0,073
Raíz del error cuadrático medio	0,200	0,198	0,344
Error absoluto relativo (%)	5,915	8,840	10,413

2.5 Conclusiones Parciales

Los métodos que se proponen en la tesis para el cálculo de reductos logran reducciones altamente significativas de la cantidad de atributos respecto al conjunto original de datos, mientras que la precisión del clasificador k-NN no se vio afectada por los conjuntos de atributos reducidos. Los algoritmos UMDA que se implementaron con las funciones de adaptabilidad reportadas por Wróblewski resultaron superiores a los algoritmos genéticos de Wróblewski, respecto a la longitud de los reductos encontrados y el tiempo de ejecución. La combinación del cálculo del núcleo con la Heurística 1, que combina criterios de ganancia y relevancia entre atributos, resultó la más conveniente para el método RSRed*, respecto al tiempo de ejecución para encontrar un reducto. El método RSRed* obtiene reductos más cortos en la mayoría de los experimentos realizados cuando se emplea la Heurística 2.

Los métodos que se proponen para la edición de conjuntos de entrenamiento: Edit1RS, Edit1RS (con reducto), Edit3RS, Edit3RS (con reducto), obtienen mejores desempeños en la clasificación y logran reducciones significativas de los objetos respecto al conjunto

original de datos. No existen diferencias significativas respecto a la reducción de objetos al usar el conjunto completo de atributos que seleccionando reducto, sin embargo, al tener menos atributos, el costo computacional es menor. Edit1RS logra retener menos objetos que Edit3RS. Si lo que se persigue con la edición es lograr mejorar la calidad de la muestra de entrenamiento para la clasificación supervisada, es conveniente editar a través del algoritmo Edit3RS. Si el objetivo principal de la edición es lograr una muestra reducida pero con buenos resultados de precisión de los clasificadores, el más conveniente en este sentido es el algoritmo Edit1RS.

Las medidas de inferencia basadas en la Teoría de los Conjuntos Aproximados: Calidad de la clasificación (medida clásica), así como las que se proponen en el presente trabajo, Precisión generalizada de la clasificación, Calidad generalizada de la clasificación y el Coeficiente de aproximación general, las cuales se refieren al desempeño de todo el sistema de decisión, están altamente correlacionadas con el desempeño general de los clasificadores supervisados Perceptron multicapa, C4.5 y k-NN. De igual forma, están altamente correlacionadas las medidas Calidad de la aproximación y Precisión de la aproximación, las cuales hacen una particularización por clases, con el desempeño por clases de los clasificadores antes mencionados.

La propuesta de construir conjuntos de datos de entrenamiento a partir de los resultados de las medidas de inferencia de la RST y los de la precisión de clasificadores supervisados (Perceptron multicapa, k-NN, C4.5), resultó exitosa para la generación de conocimiento, obteniéndose buenos desempeños en la clasificación supervisada a la hora de inferir a priori la precisión que se obtendrá con un nuevo conjunto de entrenamiento.

Todo lo anterior muestra que el empleo de la Teoría de los Conjuntos Aproximados permite la construcción de algoritmos para resolver de forma eficiente diversas tareas en el campo del aprendizaje automatizado.

Capítulo 3

Solución automatizada al problema de pronósticos de las temperaturas

3 SOLUCIÓN AUTOMATIZADA AL PROBLEMA DE PRONÓSTICO DE LAS TEMPERATURAS

La calidad y la precisión de predicciones y avisos meteorológicos se imponen para el desarrollo de cualquier país, por su utilidad en distintos sectores socioeconómicos (sector agrícola, energético, salud ciudadana, entre otros) que precisan de este tipo de predicción para poder evaluar a corto y medio plazos sus políticas de actuación ante situaciones climáticas adversas. La temperatura del aire (temperatura, como se le conoce comúnmente), como medida del contenido de calor del medio aéreo, es uno de los elementos climáticos más importantes, pues resulta un elemento indispensable para la planificación adecuada de muchas de las actividades básicas del hombre, incluyendo hasta su vestuario.

Actualmente, es posible realizar predicciones deterministas con una antelación de entre siete y diez días en las regiones extratropicales y de entre tres y cuatro, en las tropicales. Además, la predicción estacional de El Niño y La Niña, que supone otra extraordinaria innovación, es reveladora de que se ha alcanzado una mejor comprensión científica de los procesos dinámicos y físicos de la atmósfera y los océanos, y sus interacciones con otros componentes del Sistema Terrestre (Aroche 2006).

Sin embargo, el desarrollo en la predicción no es homogéneo en el mundo. Los pronósticos técnicos, durante los años 2000 al 2006, realizados por el Departamento de Pronósticos de la provincia de Camagüey, tienen un índice de eficiencia por debajo del 87,5%. Las temperaturas máximas y mínimas diarias, la nubosidad, la dirección del viento, la velocidad del viento y las precipitaciones, son las variables que se miden en la evaluación de la calidad de los pronósticos técnicos de la provincia. La variable temperatura mínima fue la de más baja calidad del pronóstico, para un promedio de efectividad de 84,65%, seguida por la variable temperatura máxima, con una efectividad del 87,25, en el período comprendido entre los años 2000 al 2006 (Aroche 2006).

No existe un algoritmo establecido para pronosticar variables meteorológicas (precipitaciones, temperaturas, nubosidad, viento, entre otras). Cada grupo de pronosticadores sigue una secuencia de pasos propia y une la información obtenida, para de

acuerdo a las características del lugar y a la experiencia individual y colectiva, dar un pronóstico de estas variables.

¿Cómo se realiza, hasta ahora, el pronóstico de las temperaturas diarias en el Centro Meteorológico de Camagüey?

Para responder esta interrogante se enuncian los pasos generales siguientes, los cuales fueron obtenidos a través de entrevistas con los especialistas pronosticadores de la provincia. Estos pasos son propios del grupo camagüeyano de pronósticos, y no están formalmente establecidos por ellos.

- Persistencia temporal de las magnitudes meteorológicas. No se deben diferenciar "mucho" (2 grados Celcius) los valores de las temperaturas del día que se pronostica y los del día anterior.
- Valores extremos de las temperaturas en el período (en el mes).
- Análisis de los factores que pudieran provocar que las temperaturas se salgan del rango preestablecido, como por ejemplo, sistemas de "altas" y "bajas".
- Peculiaridades físicas geográficas para el área que pronostica.

Para ello, los pronosticadores utilizan estadísticas de tendencia del comportamiento de las temperaturas en los siete días anteriores, valores extremos de las temperaturas en los últimos tres años, además de informaciones diarias, a las que puedan acceder, del Sistema CubaForecast y de centros meteorológicos como el Centro Europeo, Centro de Pronóstico Numérico de los Estados Unidos, Centro de Pronóstico del Caribe, entre otros. Con toda esta información cada pronosticador determina rangos y un valor determinista de las temperaturas máxima y mínima del día, lo cual se consulta dentro del grupo para dar el valor definitivo que se informa al Centro Meteorológico Nacional y a la población.

Precisamente, como ayuda a los especialistas en esta área, se ha desarrollado un sistema automatizado de predicción de temperaturas para las seis estaciones del Centro Meteorológico de Camagüey.

3.1 Descripción del problema

La provincia de Camagüey cuenta con seis estaciones meteorológicas, ubicadas en los municipios de Florida (Estación 78350), Santa Cruz del Sur (Estación 78351), Esmeralda (Estación 78352), Nuevitas (Estación 78353), Palo Seco (Estación 78354) y Camagüey (Estación 78355). El Centro Meteorológico de Camagüey, en coordinación con el Centro Nacional de Meteorología, ha facilitado los datos correspondientes a las seis estaciones meteorológicas para las variables temperaturas máximas y mínimas y otras que inciden en las variaciones de la temperatura. Se cuenta con los valores reales diarios de estos datos en el período comprendido entre los años 2000-2006, así como los valores de las temperaturas pronosticados por el Departamento de Pronósticos de Camagüey.

Las características de los atributos que describen este problema aparecen en la tabla 3.1, que se muestra a continuación.

Tabla 3.1 Descripción de los atributos del problema

Nombre	Descripción	Tipo	Ausencia de información
Año	Fecha para la cual se obtuvieron los valores de las temperaturas y las otras variables asociadas.	entero	no
Mes		entero	no
Atributos asociados a la misma variable objetivo			
$T_{x-1}, T_{x-2}, ..., T_{x-7}$	Son siete atributos diferentes, correspondientes a los valores reales de las temperaturas máximas de los siete días anteriores, se miden de las 14:00 a las 16:00 horas.	real	sí
$T_{n-1}, T_{n-2}, \ldots, T_{n-7}$	Son siete atributos diferentes, correspondientes a los valores reales de las temperaturas mínimas de los siete días anteriores, se miden de las 6:00 a las 8:00 horas.	real	sí
Atributos energéticos			
H_{luz}	Insolación	real	sí
W	Número de Wolf que se determina teniendo en cuenta el total de grupos de manchas existentes en el disco solar y el	real	sí

	total de manchas y poros que integran dichos grupos.		
Flujo solar	Informa cuantitativamente sobre la actividad solar en términos de la energía irradiada.	real	sí
Área	Área cubierta por manchas solares.	real	sí
Nuevas manchas	Número de manchas que han surgido.	entero	sí
	Atributos asociados a las condiciones del tiempo		
dd7	Dirección del viento a las 19:00 horas	entero	sí
ff7	Velocidad del viento a las 19:00 horas	entero	sí
dd00	Dirección del viento a las 7:00 horas	entero	sí
ff00	Velocidad del viento a las 7:00 horas	entero	sí
HR	Humedad relativa	entero	sí
Temperatura del día	**Atributo objetivo**. Se refiere a la temperatura máxima o mínima, en dependencia del problema. Está dado por una clase (pronóstico por rangos)	simbólico	sí

Las series de datos sobre las variables meteorológicas (atributos del problema) aparecen en la base de datos del Proyecto "Variabilidad espacio-temporal de elementos del tiempo y el clima de gran impacto socioeconómico territorial" (Aroche 2006). Los datos que se utilizaron en este trabajo, relacionados con el flujo solar, el área cubierta por manchas solares, el índice acerca de nuevas manchas y el número de Wolf, se obtuvieron de los reportes del 2007 del Sunsput Index Data Center (SIDC) de Bruselas, Bélgica y del NOAA Boulder Colorado, EE.UU.

Este es un típico problema de clasificación supervisada, donde se quiere predecir la clase correspondiente a los valores de las temperaturas máximas y mínimas, dado un nuevo objeto descrito por los atributos que aparecen en la tabla 3.1.

Los rangos establecidos por los pronosticadores para el atributo temperatura (Temp) se asocian a 13 clases, tal y como se especifica en la tabla 3.2.

Tabla 3.2 Rangos (asociados a clases) de valores para el atributo temperatura

Rangos (en grados)	**Clase**
Temp<12	A
12 ≤ Temp<16	B
16 ≤ Temp<18	C
18 ≤ Temp<20	D
20 ≤ Temp<22	E
22 ≤ Temp<24	F
24 ≤ Temp<26	G
26 ≤ Temp<28	H
28 ≤ Temp<30	I
30 ≤ Temp<32	J
32 ≤ Temp<34	K
34 ≤ Temp<36	L
Temp ≥ 36	M

Las Series Temporales en la solución del problema

Los problemas de pronósticos son resueltos frecuentemente a través de series temporales. El problema de los pronósticos de las temperaturas en el Centro Meteorológico de Camagüey se intentó resolver inicialmente a través de series de tiempo modelables ARIMA. Se trató de buscar para cada estación meteorológica un modelo para las temperaturas máximas y otro, para las mínimas. En ninguno de los casos se obtuvo un modelo válido; se hicieron varios intentos con otros tipos de modelaciones no ARIMA para los cuales tampoco se obtuvieron modelos adecuados. Esto pudo estar dado por la presencia de gran cantidad de ruidos en los datos, por lo que se realizó un preprocesamiento de los datos para eliminar sobre todo la presencia de elementos ruidosos y se intentó encontrar los modelos ARIMA con los nuevos conjuntos de datos. Sin embargo, solo se obtuvo un modelo válido para la Estación 78350 de Florida.

3.2 Solución automatizada para el pronóstico de las temperaturas

Luego de un trabajo de **recopilación** de datos por parte de los meteorólogos, se construyeron doce conjuntos de datos (para las temperaturas máximas y mínimas de cada estación), cada uno de los cuales está formado por 2556 objetos, descritos por los atributos que aparecen en la tabla 3.1.

Construcción de las bases de casos

El comportamiento de las temperaturas máximas y mínimas de un día específico está estrechamente relacionado con los siete días anteriores (Lecha and Florido 1989); por este motivo para construir las bases de casos, se tuvieron en cuenta los valores de las variables temperatura máxima y temperatura mínima de los siete días que le anteceden al que se desea pronosticar. De esta manera, se tienen 26 atributos predictores y un atributo objetivo para cada caso.

Limpieza y tratamiento de los datos

Debido a que estos conjuntos de datos de meteorología provienen de diferentes fuentes y se han presentado problemas en el almacenamiento de los mismos, existen para varios atributos, valores ausentes y valores ruidosos. Estos valores con ruido pueden provocar que el algoritmo de aprendizaje obtenga soluciones erróneas; los algoritmos de edición que se proponen en la tesis tienen implícito un tratamiento para los valores ruidosos.

Para eliminar el problema de la ausencia de información se siguieron dos estrategias (Ruiz 2006): i) eliminar el objeto completo de la base de datos y ii) sustituir el valor ausente por la media o la moda (en dependencia de si el atributo es continuo o discreto) de los valores del atributo en la clase a la que pertenece el objeto. La regla para aplicar estas estrategias es la siguiente:

- Cuando la ausencia es el valor del atributo objetivo, o cuando el número de atributos con valores ausentes del objeto, es relativamente grande (que falte el 50 % o más de los atributos predictores), se sigue la estrategia i); de lo contrario, se sigue la estrategia ii).

3.2.1 Medidas de Inferencia

En el epígrafe 2.4.3 del Capítulo 2, se mostró que existe una alta correlación entre la precisión de los clasificadores supervisados (k-NN, MLP y C4.5) y las medidas de inferencia definidas por las expresiones 1.9, 2.15, 2.16 y 2.18. Por tal motivo, primeramente se calcularon las medidas asociadas al desempeño general de la clasificación supervisada para todos los conjuntos de datos y se obtuvieron los valores que se muestran en las tablas 3.3 y 3.4.

Tabla 3.3 Resultados de las medidas de Inferencia para las temperaturas mínimas

Base de Casos	$\Gamma(DS)$	$A_G(DS)$	$\Gamma_G(DS)$	$T(DS)$
Estación 78350	0,588	0,547	0,605	0,486
Estación 78351	0,588	0,481	0,485	0,486
Estación 78352	0,773	0,571	0,757	0,581
Estación 78353	0,816	0,804	0,823	0,771
Estación 78354	0,891	0,907	0,882	0,804
Estación 78355	0,812	0,803	0,831	0,765

Tabla 3.4 Resultados de las medidas de Inferencia para las temperaturas máximas

Base de Casos	$\Gamma(DS)$	$A_G(DS)$	$\Gamma_G(DS)$	$T(DS)$
Estación 78350	0,827	0,721	0,794	0,783
Estación 78351	0,792	0,717	0,794	0,766
Estación 78352	0,792	0,717	0,794	0,766
Estación 78353	0,838	0,738	0,780	0,791
Estación 78354	0,800	0,720	0,799	0,774
Estación 78355	0,759	0,685	0,738	0,760

Al inferir a priori la precisión de los clasificadores supervisados (C4.5, MLP y k-NN), a través del método C4.5 (como se explica en el epígrafe 2.4.4 del Capítulo 2), se obtuvieron los resultados siguientes, tanto para las temperaturas mínimas, como para las máximas:

Para un C4.5 ← Aplicable, precisión baja

Para un MLP ← No aplicable, precisión muy baja

Para un k-NN ← Aplicable, precisión media

Se concluye que estos clasificadores no tendrán un buen desempeño si son entrenados con estos conjuntos de datos. Resultaría conveniente seleccionar los atributos más relevantes y realizar un proceso de edición de objetos en los conjuntos de entrenamiento, a través de los métodos que se propusieron y validaron en el Capítulo 2, con el objetivo de obtener nuevos conjuntos de entrenamiento que incrementen la precisión de los clasificadores y que permitan decrecer a su vez, el costo computacional de la clasificación supervisada.

3.2.2 Selección de atributos relevantes

Se aplicó el algoritmo RSRed* para ambas heurísticas. Con la Heurística 2, se obtienen reductos más cortos, sin embargo, con la Heurística 1 el algoritmo RSRed* consumió menos tiempo en la ejecución que con la Heurística 2. Se obtuvieron los reductos para las 6 estaciones meteorológicas. En la tabla 3.5 se muestran algunos de los reductos obtenidos.

Tabla 3.5 Algunos de los Reductos que se construyeron en el proceso de selección de atributos para las temperaturas mínima y máxima

Temperatura Mínima
Mes, temperaturas mínimas ($T_{x-1,}$ $T_{x-2,\ ...,}$ $T_{x-7)}$, W
Mes, temperaturas mínimas ($T_{x-1,}$ $T_{x-2,\ ...,}$ $T_{x-7)}$, dd00, ff00
temperaturas mínimas ($T_{x-1,}$ $T_{x-2,\ ...,}$ $T_{x-7)}$, Área, Nuevas manchas, dd7, ff7, dd00, ff00
Temperatura Máxima
Mes, temperaturas máximas ($T_{x-1,}$ $T_{x-2,\ ...,}$ $T_{x-7)}$, H_{luz}
Mes, temperaturas máximas ($T_{x-1,}$ $T_{x-2,\ ...,}$ $T_{x-7)}$, Flujo, Área, Nuevas manchas
temperaturas máximas ($T_{x-1,}$ $T_{x-2,\ ...,}$ $T_{x-7)}$, temperaturas mínimas ($T_{x-1,}$ $T_{x-2,\ ...,}$ $T_{x-7)}$, HR

3.2.3 Edición de conjuntos de entrenamiento

Los conjuntos de datos se editaron por los métodos Edit1RS y Edit3RS y se obtuvieron para cada conjunto de entrenamiento dos nuevos conjuntos editados. Los resultados respecto a la reducción de la muestra original se observan en las tablas 3.6 y 3.7.

Tabla 3.6 Resultados obtenidos en el proceso de edición de conjuntos de entrenamiento para las temperaturas mínimas

Base de Casos	% de Reducciones Edit1RS	% de Reducciones Edit3RS
Estación 78350	27,68	13,01
Estación 78351	26,45	12,83
Estación 78352	23,38	10,49
Estación 78353	28,81	13,63
Estación 78354	32,45	15,09
Estación 78355	30,23	14,25

Tabla 3.7 Resultados obtenidos en el proceso de edición de conjuntos de entrenamiento para las temperaturas máximas

Base de Casos	% de Reducciones Edit1RS	% de Reducciones Edit3RS
Estación 78350	34,53	18,05
Estación 78351	32,78	16,44
Estación 78352	29,76	13,47
Estación 78353	28,45	12,79
Estación 78354	30,06	15,03
Estación 78355	31,32	16,08

El método Edit1RS logra reducciones mayores que el Edit3RS en el conjunto de entrenamiento, sin embargo, el objetivo fundamental de aplicar el proceso de edición fue para mejorar el desempeño en el proceso de clasificación supervisada, por lo que se seleccionarán para el entrenamiento aquellos conjuntos que sean más promisorios para la obtención de una mayor precisión en la clasificación.

3.2.4 Medidas de inferencia de los conjuntos de entrenamiento preprocesados

Se calcularon las medidas de inferencia para los nuevos conjuntos de datos (editados por el método Edit3RS). Los resultados se pueden apreciar en las tablas 3.8 y 3.9.

Tabla 3.8 Resultados de las medidas de inferencia para las temperaturas mínimas con los conjuntos de datos editados por el método Edit3RS

Estación	$\Gamma(DS)$	$A_G(DS)$	$\Gamma_G(DS)$	$T(DS)$
78350	0,94	0,96	0,95	0,97
78351	0,95	0,95	0,96	0,96
78352	0,95	0,96	0,98	0,96
78353	0,96	0,96	0,98	0,97
78354	0,96	0,95	0,98	0,96
78355	0,96	0,95	0,97	0,96

Tabla 3.9 Resultados de las medidas de inferencia para las temperaturas máximas con los conjuntos de datos editados por el método Edit3RS

Estación	$\Gamma(DS)$	$A_G(DS)$	$\Gamma_G(DS)$	$T(DS)$
78350	0,97	0,95	0,95	0,96
78351	0,96	0,94	0,95	0,97
78352	0,95	0,96	0,96	0,95
78353	0,97	0,95	0,96	0,95
78354	0,96	0,96	0,95	0,94
78355	0,96	0,96	0,96	0,94

Al inferir a priori la precisión de los clasificadores supervisados (C4.5, MLP y k-NN), a través del método C4.5, con los conjuntos de entrenamiento editados, se obtuvieron los resultados siguientes, para las temperaturas mínimas y las máximas:

Para un C4.5 ← Aplicable, precisión alta

Para un MLP ← Aplicable, precisión alta

Para un k-NN ← Aplicable, precisión muy alta

De este resultado se decide que el clasificador supervisado que se utilizará para la solución del problema de los pronósticos de las temperaturas es el k-NN y se usarán como conjuntos de entrenamiento, los conjuntos editados obtenidos por el método Edit3RS.

3.2.5 Descripción del clasificador k-Vecinos más Cercanos que se usó en el sistema de pronósticos

Para la implementación del clasificador se usó el algoritmo estándar de los k-Vecinos más Cercanos (k-NN).

Un aspecto importante a tener en cuenta es el error que puede cometerse al clasificar cada objeto del conjunto de entrenamiento, este es conocido como Error de Clasificación Dejando uno Fuera (LOOCE por sus siglas en inglés). El objetivo del clasificador es minimizar el coeficiente LOOCE. La forma de calcularlo está en dependencia de si los valores de las clases son continuos o discretos.

Para valores discretos de la clase (como los del problema de pronóstico, ver tabla 3.2) se calcula como:

$$LOOCE = \sum_{q \in BC} \sum_{j \in J} \left(\delta_{q,j} - p_{q,j} \right)^2 \tag{3.1}$$

Esto significa que para cada objeto *q* que pertenece a la base de casos *BC* y para cada clase *j* que pertenece al conjunto de clases *J*, se calcula la sumatoria de la diferencia entre la función de membresía de clase ($\delta_{q,j}$) y la función de probabilidad de clase ($p_{q,j}$), definidas como:

$$\delta_{q,j} = \begin{cases} 1 & q_c = j \\ 0 & q_c \neq j \end{cases} \tag{3.2}$$

$$p_{q,j} = \frac{\sum_{r \in K} \delta_{r,j} \cdot sim(q,r)^2}{\sum_{r \in K} sim(q,r)^2} \tag{3.3}$$

Donde *K* es el conjunto de los objetos vecinos más similares.

Cuando se va a clasificar una instancia, debido a que k-NN conoce el valor de la clase de q, este devuelve la clase más probable utilizando las siguientes fórmulas:

Si los valores de la clase son discretos, k-NN devuelve:

$$q_c = \max_{j \in J} p_{q,j} \tag{3.4}$$

Donde $p_{q,j}$ está definida en la expresión 3.3.

La similitud entre dos objetos se calcula como:

$$sim(q,c) = \sum_{a=1}^{n} w_a \cdot sim_a(q_a, c_a) \tag{3.5}$$

Donde sim_a es una función de similitud utilizada para comparar el valor del atributo *a* de cada objeto, *n* la cantidad de atributos y w_a el peso del atributo *a*. k-NN le da al peso de cada atributo un valor constante:

$$\forall_{a \in A}\ w_a = s \tag{3.6}$$

Siendo *S* una constante escalar. La expresión 3.6 da igual importancia a todos los atributos por lo que en presencia de atributos redundantes, irrelevantes, interactivos o ruidosos la precisión del k-NN disminuye. En el sistema PROMETEM 1.0, esto no es un problema porque el sistema tiene un módulo para el tratamiento de los datos, que incluye tratamiento a valores ausentes, ruidosos y permite la selección de atributos relevantes.

Funciones de similitud para la comparación de atributos

Existen varias funciones de comparación de atributos (funciones de similitud), las cuales están asociadas al tipo del atributo que se compara. En (Wilson and Martínez 1997; García 2003), aparecen varias con estos propósitos, algunas de estas funciones de similitud se describen a continuación y se implementan en el sistema PROMETEM 1.0:

Similitud para tipo de dato entero y real:

Los tipos de datos entero y real son usados para almacenar números enteros y reales respectivamente. La función descrita en la expresión 3.7 se usa para estos tipos:

$$sim(x_a, q_a) = \begin{cases} 1 & si\ q_a = x_a \\ 1 - \left(\dfrac{|\, q_a - x_a \,|}{(máximo\ valor\ posible - mínimo\ valor\ posible)} \right) & eoc \end{cases} \tag{3.7}$$

Donde x_a y q_a son los valores del atributo *a* para dos objetos que se comparen, *máximo valor posible* y *mínimo valor posible* son los valores máximo y mínimo respectivamente que admite el atributo *a*.

Similitud para tipo de dato simbólico:

El tipo de dato simbólico es una enumeración arbitraria o fija de secuencias de caracteres, en presencia del mismo se puede utilizar la función definida en 3.8:

$$sim(x_a, q_a) = \begin{cases} 1 & si\ x_a = q_a \\ 0 & eoc \end{cases} \quad (3.8)$$

Donde x_a y q_a son los valores del atributo *a* para dos objetos que se comparen.

Similitud para tipo de dato booleano:

Los atributos de tipo booleano sólo pueden tomar valores “verdadero” o “falso”, la función utilizada para estos es la misma que la definida en la expresión 3.8.

Cálculo del valor para el *k* óptimo

El sistema calcula el valor de *k* que minimice el LOOCE, para ello realiza una búsqueda entre los valores del 1 al 15.

En la solución a este problema no solo se determina el valor de la clase (rango en que se encuentran las temperaturas a pronosticar), sino que se estima el valor más probable dentro de ese rango a través del estimador de funciones k-NN.

3.3 Características generales del sistema PROMETEM 1.0

PROMETEM (**Pro**nóstico **Me**teorológico de **Tem**peraturas) es una aplicación Web que cuenta con una interfaz amigable y a la vez fácil de utilizar. Está basado en la tecnología Cliente-Servidor, lo cual permite una mayor velocidad en el funcionamiento de la aplicación. Los clientes no tendrán necesidad de instalar ningún programa adicional, ni deberán contar con un procesador de alta tecnología para utilizar el sistema de manera óptima. PROMETEM fue desarrollado haciendo uso de la plataforma .NET y el lenguaje de programación de alto nivel C#.

El sistema PROMETEM 1.0 es un sistema automatizado de ayuda a los especialistas del Centro Meteorológico de Camagüey para el pronóstico de las temperaturas. El mismo permite seleccionar los atributos y objetos más relevantes de un conjunto de datos que se usará para el entrenamiento; entrenarse, a través de los datos y realizar los pronósticos de las temperaturas máximas y mínimas para una estación dada; almacenar el conocimiento adquirido, así como los pronósticos realizados en un histórico; graficar los pronósticos realizados en mapas geográficos: de las estaciones, de la provincia, del país; así como graficar resultados estadísticos de tendencia respecto a la evolución temporal del pronóstico y la temperatura real.

El sistema cuenta con cuatro módulos fundamentales:

Módulo de Codificadores. A través de este módulo se podrán manipular los listados de estaciones, mapas y relación de ubicación que existe entre ellos con el fin de realizar la representación gráfica de los pronósticos de temperaturas (ver figura 3.1).

Figura 3.1 Manipulación en el sistema de estaciones y mapas asociados

Módulo de entrenamiento y adquisición del conocimiento. A través de este módulo la aplicación será capaz de adquirir conocimiento a partir de bases de casos relacionadas con el pronóstico de temperaturas, así como depurar (a través de la edición de los conjuntos de entrenamiento y la selección de atributos relevantes) el contenido de estas en aras de mejorar la calidad de los pronósticos. En este módulo se manipula el listado de

entrenamientos almacenados en la base de datos del sistema. Es posible seleccionar la estación que estará asociada al listado de entrenamientos para una mejor manipulación de los mismos, agregar y eliminar aquellos que el especialista desee.

El sistema deberá adquirir el conocimiento a partir de un fichero con un formato específico (el de fichero de datos). Una vez concluido el proceso de entrenamiento, se muestran los gráficos que indican la calidad del conocimiento adquirido. La figura 3.2 es un ejemplo de uno de los entrenamientos que se realizó con el sistema. En el gráfico de error por objeto se muestra el error cometido al clasificar cada una de las instancias en la base de casos, luego de realizar el entrenamiento.

Figura 3.2 Gráfico de error de clasificación por objetos

En el gráfico de pesado de rasgos se muestra el nivel de importancia en forma de peso que el sistema le asignó a cada atributo (rasgo) en el proceso de entrenamiento. Un ejemplo se muestra en la figura 3.3.

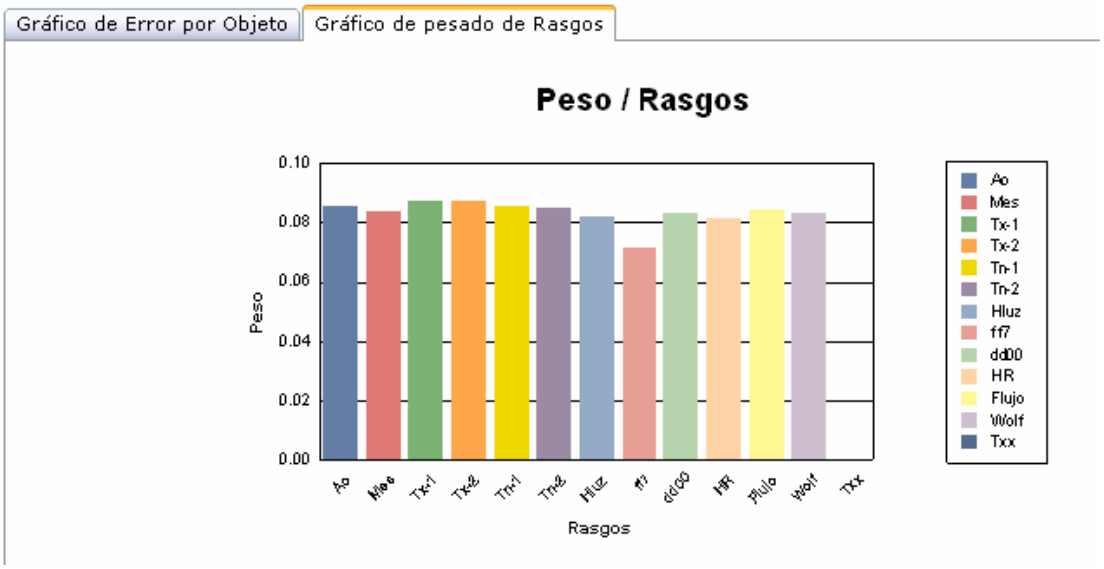

Figura 3.3 Gráfico de pesado de rasgos

Módulo de pronósticos de temperaturas. En este módulo se realizará el pronóstico de temperaturas a partir del conocimiento adquirido para luego almacenarlo, en caso de ser deseado, en la base de datos de la aplicación (ver figura 3.4).

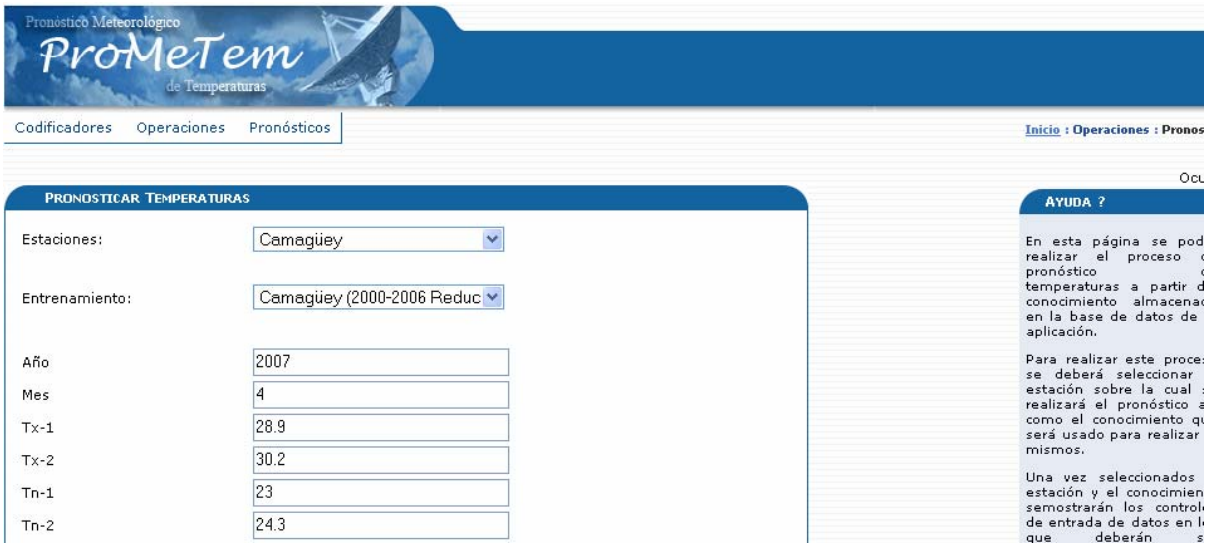

Figura 3.4 Gráfico de pesado de rasgos

Módulo de graficación de temperaturas pronosticadas. A través de este módulo se podrán representar cada uno de los pronósticos realizados sobre los mapas asociados a las

estaciones registradas en la aplicación. Es posible visualizar los pronósticos de temperaturas realizados por el sistema en una fecha determinada y sobre un mapa específico. Para graficar un pronóstico se deberá indicar la fecha en la que se realizó el pronóstico en "Seleccione la fecha" así como el mapa sobre el que se graficará en "Seleccione el mapa" y por último hacer clic en el botón "Ver pronóstico". Los pronósticos se grafican sobre el mapa en forma de círculo y al mover el Mouse por encima de estos, se visualizan los detalles de cada pronóstico, como se muestra en la figura 3.5.

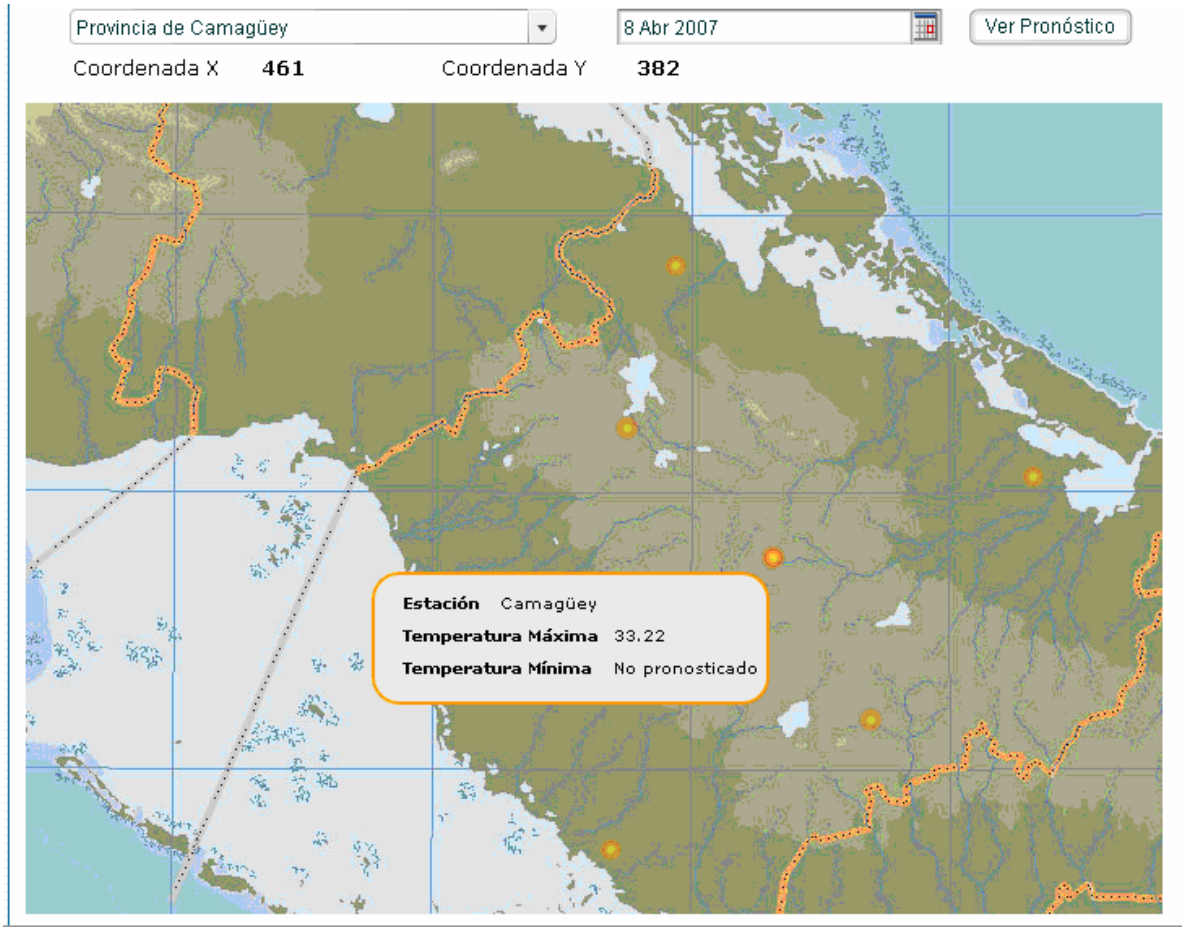

Figura 3.5 Pronóstico visual en un mapa de Camagüey

3.3.1 Aspectos fundamentales para el trabajo con el sistema PROMETEM 1.0

Esta herramienta utiliza dos ficheros fundamentales, uno de inicialización que contiene la información necesaria para inicializar el razonador de la aplicación, y el otro es el fichero que contiene los ejemplos de entrenamiento.

Formato del fichero de inicialización

El fichero de inicialización es un fichero de texto con extensión (*.hd). Cada línea contiene la información de los atributos que describen los ejemplos (instancias), por lo que cada línea en este fichero deberá aparecer en el mismo orden que aparecen los atributos en el fichero de ejemplos.

El formato de las líneas que deben añadirse para describir todos los tipos de atributos que acepta el sistema se indica a continuación.

Para atributos de tipo entero debe colocarse:

INTEGER, <Nombre del Atributo>, <mínimo valor posible>, <máximo valor posible>

Ej. *INTEGER, Wolf, 0, 352*

Para atributos de tipo real debe colocarse:

REAL, <Nombre del Atributo>, <mínimo valor posible>, <máximo valor posible>

Ej. *REAL, Txx, 3.3, 31.9*

Aunque los atributos del problema son todos enteros o reales, el sistema no se limita a aceptar solamente estos tipos de atributos, pues en un futuro pudieran considerarse otras variables para el pronóstico meteorológico que no tienen por qué ser de la misma naturaleza que la de los datos actuales, como por ejemplo: atributos de tipo difuso, booleano, simbólico, símbolos ordenados y conjunto de símbolos.

Opcionalmente se puede colocar en el fichero de inicialización el nombre del fichero de ejemplos, para ello debe colocarse una línea, en cualquier parte del fichero con el formato:

Name, <Nombre del fichero de ejemplos>

Ej. *Name, Camagüey_Máx*

Formato de fichero de datos (conjunto de objetos)

El fichero de datos no es más que un fichero de texto con extensión (*.data), en el cual cada línea contiene la lista de valores de los atributos, separados por coma, los cuales corresponden a la descripción de un objeto del conjunto de datos.

Para representar los valores de tipo conjunto se colocan los valores que contiene el conjunto separados por el caracter "|". Ej. 3.3, *1|2*, 10

Estructura de las páginas del sistema

Para una mejor familiarización y uso de la aplicación por parte de los usuarios, la mayoría de las páginas del sitio cuentan con una estructura común:

- Barra de menús: se muestran las distintas opciones del menú de la aplicación.
- Encabezado del contenido: en esta área se muestra un texto referente al contenido operacional de la página.
- Área de trabajo: en esta región estará contenida el área de trabajo de las diferentes páginas que conforman la aplicación.
- Ayuda: ayuda referente al uso operacional de la página en la que se encuentre navegando el usuario.
- Barra de navegación: muestra la ubicación en la que se encuentra el usuario en la aplicación y permite navegar a las diferentes páginas que en esta se muestran.

Descripción de los menús de la aplicación

A continuación se explica brevemente algunas de las opciones que se muestran en la barra de menús de la aplicación (figuras 3.6 y 3.7).

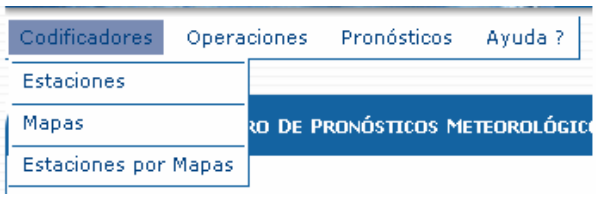

Figura 3.6 Menú "Codificadores" del sistema

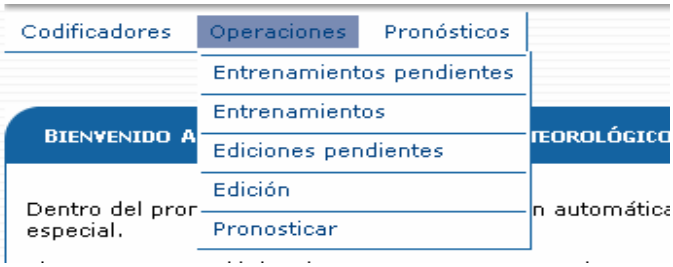

Figura 3.7 Menú "Operaciones" del sistema

Estaciones: permite acceder a la página en la que se podrá manipular el listado de estaciones de las que se pronosticarán las temperaturas.

Mapas: permite acceder a la página en la que se podrá manipular el listado de mapas sobre los que se mostrará, de forma gráfica, el pronóstico de temperaturas de las distintas estaciones.

Estaciones por Mapas: permite establecer relaciones de ubicación geográfica entre los diferentes mapas y estaciones registradas en la base de datos de la aplicación.

Entrenamientos: permite acceder a la página en la que se podrá manipular el listado de entrenamientos realizados con anterioridad así como realizar nuevos entrenamientos para la adquisición de conocimiento por parte del sistema.

Ediciones pendientes: permite acceder a la página donde se podrá manipular el listado de tareas de edición que se están realizando en estos momentos.

Edición: permite acceder a la página en la que se podrán realizar nuevas ediciones, seleccionando uno de los métodos de edición del sistema (Edit1RS, Edit3RS) para preprocesar el conjunto de datos que se utilizará para el entrenamiento. Aquí además se puede seleccionar un conjunto de atributos relevantes (método RSRed*) para realizar la edición con un reducto.

Pronosticar: permite acceder a la página en la que se podrá realizar el pronóstico de temperaturas, así como almacenar el mismo en la base de datos de la aplicación. Es posible realizar el pronóstico diario de las temperaturas y el pronóstico de los próximos diez días. Además de dar la clase (el pronóstico por rangos), se especifica el valor más apropiado dentro de la clase, lo cual se realiza a través del clasificador k-NN que se usó en el sistema.

3.4 Resultados obtenidos por el sistema de pronósticos PROMETEM 1.0

Se realizaron varias pruebas para comprobar el desempeño del sistema PROMETEM 1.0. Se tomó cada uno de los años del estudio como conjunto control y se compararon los resultados del sistema con los resultados que obtuvieron los pronosticadores del grupo de Camagüey, para ese año. Además, se tomó el conjunto de control de forma totalmente aleatoria y se repitió este proceso diez veces, para realizar una prueba de validación cruzada. El resultado promedio de las diez particiones fue de 97,78 para las temperaturas mínimas y de 98,03 para las temperaturas máximas.

La precisión obtenida por el sistema PROMETEM 1.0 al tomar como conjunto de control cada año y los otros como conjunto de entrenamiento, para el período del 2000 al 2006, se muestra en la tabla 3.10; mientras que las obtenidas por los pronosticadores en esos años son las que aparecen en la tabla 3.11.

En todos los casos de prueba, los porcientos de precisión del sistema superaron las estadísticas de los pronosticadores.

Tabla 3.10 Resultados promedio de las precisiones del sistema PROMETEM para las seis estaciones meteorológicas de Camagüey, para los años 2000 al 2006

Año	Temperaturas mínimas	Temperaturas máximas
2000	97,94	98,02
2001	98,82	98,97
2002	99,25	99,07
2003	97,47	98,87
2004	98,88	99,09
2005	98,56	98,53
2006	98,03	98,68

Tabla 3.11 Promedio de resultados de la predicción que hicieron los pronosticadores para las seis estaciones meteorológicas de Camagüey, para los años 2000 a 2006

Año	Temperaturas mínimas	Temperaturas máximas
2000	70.80	69.10
2001	81.90	89.60
2002	85.70	88.90
2003	88.30	90.90
2004	85.10	90.10
2005	92.40	91.30
2006	88.40	90.90

La tabla 3.12 muestra los resultados del sistema de pronóstico de las temperaturas del 27 de marzo al 2 de abril del 2007, para la estación Camagüey.

Tabla 3.12. Resultados del sistema PROMETEM 1.0 en el pronóstico de las temperaturas del 27 de marzo al 2 de abril del 2007, para la estación Camagüey

Día	27	28	29	30	31	1	2
Clase	G	H	H	I	I	I	I
Valor pronosticado	24.9	28.0	28.0	28.3	28.1	28.0	28.1
Valor real	25.3	27.5	27.8	28.8	28.7	28.5	28.7

Esta tabla muestra 100% de aciertos por clases (pronóstico por rango) y un error de 0.47 como promedio en el pronóstico del valor más probable (pronóstico determinista). Sin embargo, teniendo en cuenta el criterio que se sigue para la evaluación del pronóstico de la temperatura, el sistema fue acertado también para el 100 % de los días en el pronóstico determinista: *Un pronóstico se considera acertado cuando el valor real no se aleja más de 2 grados por encima o por debajo del valor pronosticado*.

3.5 Conclusiones Parciales

En la solución al problema de pronósticos automatizados de las temperaturas máximas y mínimas para el Centro Meteorológico de Camagüey fue necesario hacer un preprocesamiento de los datos antes del proceso de entrenamiento del clasificador supervisado. Esto estuvo dado por el hecho de que al estimar a priori la calidad de los

conjuntos de entrenamiento, indicaba que no se iban a obtener porcientos altos de precisión con los conjuntos de entrenamiento que se tenían.

Se seleccionaron conjuntos de atributos relevantes para las seis estaciones meteorológicas, a través del método RSRed*, con las dos heurísticas propuestas. La Heurística 2 obtuvo reductos más cortos en la mayoría de los casos, sin embargo, la Heurística 1 construyó los reductos en menor tiempo. Se construyeron conjuntos de entrenamiento editados a partir de los resultados obtenidos por el método Edit3RS, pues fue el que construyó muestras de entrenamiento con mejor calidad para el desempeño del clasificador, a pesar de que este método Edit3RS no redujo los conjuntos de entrenamiento tanto como el Edit1RS.

Se decidió utilizar el clasificador supervisado k-NN pues con todos los conjuntos de entrenamiento, el método C4.5, utilizado para inferir conocimiento a priori acerca del desempeño del proceso de clasificación, arrojó que el clasificador k-NN era aplicable, y que se obtendrían resultados muy altos de precisión general.

El sistema automatizado para el pronóstico de las temperaturas obtuvo resultados superiores a los que obtuvieron los pronosticadores en los años 2000 al 2006.

CONCLUSIONES

- Los métodos que se proponen en la tesis para el cálculo de reductos logran reducciones altamente significativas de la cantidad de atributos respecto al conjunto original de datos, mientras que la precisión del clasificador k-NN no se vio afectada por los conjuntos de atributos reducidos. La utilización de las funciones descritas por Wróblewski (Wróblewski 1995) en los Algoritmos UMDA resultó exitosa; estos superaron a los algoritmos genéticos de Wróblewski y a la mayoría de los métodos con los cuales se comparó, respecto a la longitud de los reductos encontrados y al tiempo de ejecución. Los algoritmos UMDA+f1 y UMDA+f3 realizan el cálculo de los reductos cortos en tiempos pequeños cuando el conjunto de ejemplos no es muy grande (<5000 objetos), aún cuando el número de atributos que describe el problema sí lo es. La mejor combinación resultó con la función *f*1, respecto al tiempo de ejecución; sin embargo, *f*3 logró encontrar mayor número de reductos en tiempos aceptables.

- El método propuesto RSRed* con la variante de la Heurística 2 construyó reductos más cortos que los métodos con los cuales se comparó, y en la mayoría de los casos RSRed* con la Heurística 1, consumió menos tiempo para el cálculo del reducto respecto a otros métodos reportados, mientras que la longitud del reducto obtenido no fue significativamente diferente, respecto a estos.

- Con los conjuntos de entrenamiento editados por los métodos propuestos: Edit1RS, Edit3RS (con reducto y todos los atributos) se lograron resultados de desempeño superiores en la clasificación supervisada que con los otros métodos de edición con los cuales se comparó. Estos métodos propuestos logran reducciones significativas de los objetos respecto al conjunto original de datos. No existen diferencias significativas respecto a la reducción de los objetos de la muestra usando el conjunto completo de atributos que seleccionando reducto. Si el objetivo fundamental de la edición es encaminado hacia las reducciones que se logran respecto al conjunto original y además con esta muestra editada se quiere obtener buen desempeño de los clasificadores, es conveniente editar por el algoritmo Edit1RS. Sin embargo, si lo que se persigue es lograr una muestra editada que logre

mejor desempeño en la clasificación, sería más conveniente editar la muestra con el algoritmo Edit3RS.

- Las nuevas medidas de inferencia propuestas, basadas en la Teoría de los Conjuntos Aproximados, están altamente correlacionadas con la precisión de clasificadores supervisados (Perceptron multicapa, k-NN y C4.5). Estas medidas de la RST se emplearon como atributos para formar conjuntos de datos de entrenamiento que fueron utilizados en la generación de conocimiento. Con el generador de reglas C4.5 se predice la precisión de estos clasificadores en términos de "No aplicable, precisión muy baja"; "Aplicable, precisión baja"; "Aplicable, precisión media"; "Aplicable, precisión alta" y "Aplicable, precisión muy alta", con mejores desempeños que al predecirlo con la red neuronal Perceptron multicapa. Con el método de estimación de los Vecinos más Cercanos se calcula el valor numérico de la precisión del clasificador que resultó más conveniente según el generador de reglas C4.5. Esta propuesta de inferir a priori el clasificador más conveniente y la precisión que se obtendrá para un nuevo conjunto de entrenamiento, resultó satisfactoria.
- En la solución al problema del pronóstico automatizado de las temperaturas máximas y mínimas del Centro Meteorológico de Camagüey, se realizó un preprocesamiento de los datos para seleccionar los atributos relevantes y editar los conjuntos de entrenamiento. Se determinó a priori que el clasificador más conveniente para solucionar este problema sería un k-NN y que con éste se podrían obtener resultados de precisión elevados, lo cual se pudo corroborar. El empleo de los algoritmos desarrollados mejoró significativamente el resultado alcanzado respecto al método tradicional.

RECOMENDACIONES

- Evaluar la efectividad de los algoritmos desarrollados (en la selección de atributos relevantes y la edición de conjuntos de entrenamiento) para el caso de otros métodos de aprendizaje.

- En la construcción de la propuesta para inferir a priori el desempeño de los clasificadores a partir de las medidas de inferencia de la Teoría de los Conjuntos Aproximados, se recomienda aplicar la teoría de la lógica difusa para categorizar la precisión de los clasificadores en lugar de aplicar el método de discretización por percentiles. De esta manera, se pueden construir funciones de pertenencia a cada una de las categorías (precisión muy baja, precisión baja, precisión media, precisión alta y precisión muy alta) y tener un atributo objetivo difuso.

REFERENCIAS BIBLIOGRÁFICAS

Aha, D. (1992). "Tolerating noisy, irrelevant and novel attributes in instancebased learning algorithms" Computational Journal of Man-Machine Studies(36): 267-287.

Aha, D., D. Kibler, et al. (1991). InstanceBased Learning Algorithms.

Ahn, B. S. (2000). The integrated methodology of rough set theory and artificial neural networks for business failure predictions

Ainslie, M. C. and J. S. Sánchez (2002). Space Partitioning for Instance Reduction in Lazy Learning Algorithms.

Alpigini, J. F., J. Peters, et al. (2002). Rough sets and current trends in computing Third International Conference, RSCTC 2002, Malvern, PA, USA., Lectures Notes in Computer Science 2475 Springer 2002

Arco, L., R. Bello, et al. (2006). On clustering validity measures and the Rough Set Theory. 5th Mexican International Conference on Artificial Intelligence, IEEE Computer Society Press

Aroche, R. (2006). Variabilidad espacio-temporal de elementos del tiempo y el clima de gran impacto socioeconómico territorial. Camagüey, Cuba, Centro Meteorológico.

Barandela, R., E. Gasca, et al. (2002). "Correcting the Training Data." Pattern Recognition and String Matching.

Bazan, J., N. H. Son, et al. (2003). A View on Rough Set Concept Approximations. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Bazan, J. G. and M. Szczuka (2005). The Rough Set Exploration System. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Bell, D. and J. Guan (1998). "Computational methods for rough classification and discovery" ASIS **5**(49): 403-414.

Bello, R., A. Nowe, et al. (2005)a. A model based on Ant Colony System and Rough Set Theory to Feature Selection. Genetic and Evolutionary Conference (GECCO05), Washington, USA.

Bello, R., A. Nowe, et al. (2005)b. Using ACO and Rough Set Theory to Feature Selection. 6th WSEAS Evolutionary Computing Conference (EC05), Lisbon, Portugal.

Bello, R., A. Nowe, et al. (2006). "Two Step Ant Colony System to Solve the Feature Selection Problem" Lectures Notes on Computer Sciences. Springer-Verlag: 588-596.

Beynon, M. J. (2003). Degree of Dependency and Quality of Classification in the Extended Variable Precision Rough Sets Model. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Bezdek, J. C. and L. I. Kuncheva (2004). "Nearest Prototype classifiers design: an experimental study."

Borgelt, C. and R. Kruse (2003). Neural Networks, Working Group Neural Networks and Fuzzy Systems. Otto-von-Guericke. University of Magdeburg.

Borracci, R. A. and E. B. Arribalzaga (2005). "Aplicación de análisis de conglomerados y redes neuronales artificiales para la clasificación y selección de candidatos a residencias médicas." Educación Médica **8**(1).

Bosc, P. and H. Prade (1993). An introduction to fuzzy set and possibility theory based approaches to the treatment of uncertainty and imprecision in database management system. Proc. of Second Workshop Uncertainty management, Information Systems: from Needs to Solution, California.

Brighton, H. and C. Mellish (2002). Advances in Instance Selection for Instance-Based Learning Algorithms. Data Mining and Knowledge Discovery. **6:** 53-172.

Caballero, Y. (2005). Uso de los Conjuntos Aproximados para el tratamiento de los datos. Santa Clara, Cuba, Universidad Central de Las Villas.

Caballero, Y., L. Arco, et al. (2007). New Measures for Evaluating Decision Systems using Rough Set Theory: The Application in Seasonal Weather Forecasting. Third International ICSC Symposium on Information Technologies in Environmental Engineering (ITEE'07), Carl von Ossietzky Universität Oldenburg. Alemania, Springer Verlag.

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

Caballero, Y., R. Bello, et al. (2004). Estudio de conjuntos de entrenamientos para redes tipo MLP usando medidas de la teoría de los conjuntos aproximados. I Simposio Iberoamericano de Inteligencia Artificial dentro de Informática'2004, La Habana.

Caballero, Y., R. Bello, et al. (2006)b. Improving the k-NN method: Rough Set in edit training set. The First IFIP International Conference on Artificial Intelligence in Theory and Practice, Springer Boston.

Caballero, Y., R. Bello, et al. (2007). La Teoría de los Conjuntos Aproximados en el mejoramiento de los conjuntos de entrenamiento en Bioinformática. II Congreso Internacional de Bioinformática y Neuroinformática. Informática 2007, La Habana, Cuba.

Cabello, J. T., M. B. Castello, et al. (2005). "Redes neuronales artificiales en Medicina Intensiva. Ejemplo de aplicación con las variables del MPM II." Medicina Intensiva **29**(1): 13 - 20.

Calderón, G. S. and P. P. Jara (2006). "Estudio de series temporales de contaminación ambiental mediante técnicas de redes neuronales artificiales." Revista chilena de ingeniería **14**(3): 284-290.

Carlin, U. S. (1998). Rough set analysis of medical datasets and A case of patient with suspected acute appendicitis ECAI 98 Workshop on Intelligent data analysis in medicine and pharmacology.

Carrasco, J. A., J. Ruíz, et al. (2004). Feature Selection Using Typical e -Testors, Working on Dynamical Data. Progress in Pattern Recognition, Image Analysis and Applications, Puebla, Mexico, Springer Verlag.

Cofiño, A. (2004). Técnicas estadísticas y neuronales de agrupamiento adaptativo para la predicción de fenómenos meteorológicos locales. Aplicación en el corto plazo y en la predicción estacional. Grupo de investigación AIMET. Santander, España, Universidad de Cantabria.

Cortijo, F. J. and N. Pérez de la Blanca (1997). "A comparative study of some non-parametric spectral classifiers. Applications to problems with high-overlapping training sets " International Journal of remote Sensing **18**(6): 1259-1275.

Cotta, C. and P. Moscato (2004). Evolutionary Computation: Challenges and Duties. Frontiers of Evolutionary Computation. A. Menon. Boston MA, Kluwer Academic Publishers**:** 53-72.

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

Cheguis, I. A. and S. V. Yablonskii (1958). Métodos lógicos para control del trabajo de esquemas eléctricos. Instituto de Matemática “V. A. Steklov”. URSS. **51** 236-269.

Chin, K. S., J. Liang, et al. (2003). Rough Set Data Analysis Algorithms for Incomplete Information Systems. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Choubey, S. K. (1996). A comparison of feature selection algorithms in the context of rough classifiers Fifth IEEE International Conference on Fuzzy Systems.

Chouchoulas, A. (1999). A Rough Set Approach to Text Classification.

Chouchoulas, A. and Q. Shen (1999). "A rough set-based approach to text classification " Lectures Notes in Artificial Intelligence **11**(17): 118-127.

Dasarathy, B. V., J. S. Sanchez, et al. (2000). "Nearest Neighbour Editing and Condensing Tools - Synergy Exploitation." Pattern Analysis & Applications.

Dash, M. and H. Liu (2003). Consistency-based search in feature selection. Artificial intelligence**:** 155-176.

Demsar, J. (2006). "Statistical comparisons of classifiers over multiple data sets." Journal of Machine Learning Research(7): 1-30.

Deogun, J. S. (1995). Exploiting upper approximations in the rough set methodology. First International Conference on Knowledge Discovery and Data Mining, Canada.

Deogun, J. S. (1998). "Feature selection and effective classifiers." Journal of ASIS **49**(5): 423-434.

Devijver, P. A. and J. V. Kittler (1980). On the edited nearest neighbor rule. 5th International Conference on Pattern Recognition, Los Alamitos, California.

Díaz, Y. and A. Fernández (2003). CubaForecast para el pronóstico de variables meteorológicas. C. M. d. Cienfuegos. Cuba. **3.8**.

Dimitriev, A. N., J. I. Zhuravlev, et al. (1966). "Acerca de los principios matemáticos de la clasificación de objetos y fenómenos." Diskretnyi Analiz 7: 3-15.

Ding, C. and H. Peng (2003). "Minimum redundancy feature selection from microarray gene expression data." IEEE Computer Society Bioinformatics: 523-529.

Djouadi, A. (1990). "The quality of training sets estimates of the Bhattacharyya Coefficient." Trans. On Pattern Recognition analysis and Machine learning **12**(1): 92-97.

Duarte, L. (2003). Biblioteca para el desarrollo de algoritmos evolutivos que estiman distribuciones. Santa Clara, Cuba, Universidad Central “Marta Abreu” de Las Villas.

Dunstsh, I. and G. Gunter. (2000). "Rough set data analysis." from http://citeseer.nj.nec.com/dntsch00rough.html

Elwyn, G., A. Edwards, et al. (2001). "Decision analysis in patient care." The Lancet **358**(9281): 571-574.

Feng, C. and D. Michie (1994). Statlog's ML Algorithm. Machine Learning, Neural and Statistical Classification. D. Michie, D. J. Spiegelhalter and C. C. Taylor**:** 65-77.

Ferri, C., J. Hernández, et al. (2004). Un sistema para la extracción de conocimiento en Bioinformática. IBERAMIA2004.

Fix, E. and J. Hodges (1951). "Discriminatory analysis - Nonparametric discrimination: Consistency properties." USAF School of Aviation Medicine. Randolph Field, Texas, Tech. Report **4**(4).

Freeman, J. A. and D. M. Skapura (1991). Neural Networks. Algorithms, Applications and Programming Techniques, Addison-Wesley.

García, J. M. (2003). KNN Workshop. Suite para el Desarrollo de Clasificadores Basados en Instancias. Departamento Computación. Facultad de Matemática, Física y Computación, Universidad Central “Marta Abreu” de Las Villas.

García, J. M., C. Vidal, et al. (2004). "Estudio de tumores de estirpe vascular mediante técnicas de reconocimiento de patrones." Reconocimiento de Patrones.

García, M. and J. Shulcloper (2005). "Selecting Prototypes in Mixed Incomplete Data." Lectures Notes in computer Science (LNCS 3773): 450-460.

García, M., Y. Villuendas, et al. (2005). Métodos de selección y construcción de objetos para el mejoramiento de un clasificador supervisado: estado del arte.

García, N., M. Gámez, et al. (2006). RNA+SIG: Sistema automático de valoración de viviendas. Campus Multidisciplinar en Percepción e Inteligencia por los 50 Años de la Inteligencia Artificial, CMPI-2006, Albacete, España.

Gates, G. W. (1972). "The reduced nearest neighbor rule." IEEE Trans. on Information Theory **431-433**.

Gomolinska, A. (2005). Rough Validity, Confidence, and Coverage of Rules in Approximation Spaces. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

González, P. and B. Cases (2006). TOPOS: Reconocimiento de patrones temporales en sonidos reales con redes neuronales de pulsos. Campus Multidisciplinar en Percepción e Inteligencia por los 50 Años de la Inteligencia Artificial, CMPI-2006, Albacete, España.

Grabowski, A. (2003). "Basic Properties of Rough Sets and Rough Membership Function." Journal of Formalized Mathematics **15**.

Greco, S. (2001). "Rough sets theory for multicriteria decision analysis." European Journal of Operational Research 129: 1-47.

Greco, S., M. Inuiguchi, et al. (2003). Rough Sets and Gradual Decision Rules. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Grzymala-Busse, J. W. (1994). Managing uncertainty in machine learning from examples. Proceedings of the Workshop Intelligent Information System III. Polonia.

Grzymala-Busse, J. W. and S. Siddhaye (2004). Rough set approaches to rule induction from incomplete data. 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Bases systems IPMU2004, Perugia, Italy.

Guevara, M. E. (2004). Algoritmo Genético para la selección de variables y cálculo de su importancia informacional, usando el concepto de FS-Testore Difuso. Benemérita, Universidad Autónoma de Puebla.

Gutiérrez, I. (2003). Un Modelo para la Toma de Decisiones usando Razonamiento Basado en Casos en condiciones de Incertidumbre. Departamento de Ciencias de la Computación, Facultad de Matemática-Física y Computación. Universidad Central "Marta Abreu" de las Villas. Santa Clara, Cuba.

Guyon, I. and A. Elissee (2003). "An introduction to variable and feature selection." Machine Learning Research **3**: 1157-1182.

Hall, M. A. (1998). Correlation-based Feature Subset Selection for Machine Learning, University of Waikato.

Hamamoto, Y., S. Uchimura, et al. (1997). "A bootstrap technique for nearest neighbor classifier design." IEEE transactions on pattern analysis and Machine intelligence **19**(1): 73-79.

Hart, P. E. (1968). "The Condensed Nearest Neighbor Rule." IEEE Transactions on Information Theory **14**: 515-516.

Herrera, F. (2004). Sistemas difusos evolutivos. XII Congreso Español sobre Tecnlogías y Lógica Fuzzy, Universidad de Jaén.

Hor, C.-L. and P. A. Crossley (2005). Knowledge Extraction from Intelligent Electronic Devices. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Hu, X. T., T. Y. Lin, et al. (2003). A New Rough Sets Model Based on Database Systems. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Jensen, R. and S. Qiang (2003). Finding rough sets reducts with Ant colony optimization. UK Workshop on Computational Intelligence.

Jeon, G. and J. Jeong (2006). Designing a video deinterlacing system based on Rough Set attributes reduction. The International Symposium on Fuzzy and Rough Sets. ISFUROS2006, Santa Clara, Cuba.

Jiang, Y. and Z. Zhou (2004). Editing training data for kNN classifiers with neural network ensemble. Advances in Neural Networks. H. Springer-Verlag, LNCS 3173**:** 356-361.

Kelly Christian, C. D., Higgins Robert (2001). Case-Based Reasoning.

Kibler, D. and D. Aha (1987). Learning representative exemplars of concepts: An initial case study. Fourth International Workshop on Machine Learning, Irvine, CA.

Kierczak, M., W. R. Rudnicki, et al. (2006). Construction of rough sets-based classifiers for predicting HIV resistance to nucleoside reverse transcriptase inhibitors. The International Symposium on Fuzzy and Rough Sets. ISFUROS2006, Santa Clara, Cuba.

Klose, A. A. (2003). Partially Supervised Learning of Fuzzy Classification Rules, Facultat fur Informatik. Otto-von-Guericke. Magdeburg**:** 180.

Koczkodaj, W. W. (1998). "Myths about Rough Set Theory." ACM **41**(11).

Koggalage, R. and S. Halgamuge (2004). "Reducing the Number of Training Samples for Fast Support Vector Machine Classification." **2**(3).

Kohavi, R. and B. Frasca (1994). Useful Feature Subsets and Rough Set Reducts. Third International Workshop on Rough Sets and Soft Computing.

Kohonen, T. (1995). Self - Organizing Maps, Germany: Springer.

Komorowski, J. and Z. Pawlak (1999). "Rough Sets: A tutorial." Rough Fuzzy Hybridization: A new trend in decision-making. Springer: 3-98.

Koplowitz, J. and T. A. Brown (1978). On the relation of performance to editing in nearest neighbor rule. 4th International Joint Conference on Pattern Recognition, Japan.

Korzes, M. and S. Jaroszewicz (2005). Finding reducts without building the discernibility matrix. 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Wroclaw, Poland, IEEE Computer Society.

Kostek, B., P. Szczuko, et al. (2005). Processing of Musical Data Employing Rough Sets and Artificial Neural Networks. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Kuncheva, L. I. (2004). Combining pattern classifiers. Methods and algorithms, John Wiley & Sons.

Kuo, T.-F. and Y. Yajima (2003). Approximate Reducts of an Information System. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Larrañaga, P. and J. A. Lozano (2001). Estimation of Distributions Algorithms. A New Tool for Evolutionary Computation, Kluwer Academic Publishers.

Lazo, M., J. Ruiz, et al. (2001). "An overview of the evolution of the concept of testor." Pattern Recognition(34): 753-762.

Lecha, L. and A. Florido (1989). Principales características climáticas del régimen térmico del archipiélago cubano. La Habana, Cuba.

Lee, S., N. Propes, et al. (2002). Rough Set Feature Selection and Diagnostic Rule Generation for Industrial Applications. Third International Conference, RSCTC 2002 Malvern, PA, USA, LNCS24752002 Rough Sets and Current Trends in Computing.

Liu, H. and L. Yu (2005). "Toward integrating feature selection algorithms for classification and clustering." IEEE Trans. On knowledge and data enginnering. **17**(3): 1-12.

Lowe, D. (1995). Similarity Metric Learning for a VariableKernel Classifier. Neural Computation**:** 7285.

Lozano, M., J. S. Sánchez, et al. (2003). "Training Set Size Reduction by Replacing Neighbouring Prototypes."

Malinen, J., M. Maltamo, et al. (2003). "Predicting the internal quality and value of Norway spruce trees by using two non-parametric nearest neighbor methods." Forest Products journal **53**(4): 85-94.

Martínez, F., G. Sánchez, et al. (2002). "GeneticAlgorithm to Compute FS-Testors." WESEAS Transaction on Systems **1**(2): 267-272.

Mi, J.-S., W.-Z. Wu, et al. (2003). Approaches to Approximation Reducts in Inconsistent Decision Tables. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing 9th International Conference, RSFDGRC2003, Chongqing, China.

Miao, D. and L. Hou (2003). An Application of Rough Sets to Monk's Problems Solving. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Michie, D., D. J. Spiegelhalter, et al. (1994). The Statlog Project. Machine Learning, Neural and Statistical Classification. D. Michie, D. J. Spiegelhalter and C. C. Taylor**:** 4.

Midelfart, H., J. Komorowski, et al. (2003). "Learning rough set classifiers from gene expression and clinical data." Fundamenta Informaticae **53**: 155-183.

Milone, D. H. (2005). "Reconocimiento automático del habla con redes neuronales artificiales." Ciencia, Docencia y Tecnología(31): 261-322.

Mitchell, T. (1997). Machine Learning, McGraw Hill.

Mitra, S. (2005). Computational Intelligence in Bioinformatics. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Moshkov, M. J. (2005). Time Complexity of Decision Trees. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Mühlenbein, H. (1998). "The equation for the response to selection and its use for prediction " Evolutionary Computation **5**(3): 303-346.

Mühlenbein, H., T. Mahnig, et al. (1999). "Schemata, distributions and graphical models on evolutionary optimization." Journal of Heuristics **5**(2): 215-247.

Odorico, R. (1997). "Learning vector quantization with training counters (LVQTC)." Neural Newtorks **10**(6): 1083-1088.

Ohrn, A., J. Komorowski, et al. (1998). "The Design and Implementation of a Knowledge Discovery Toolkit Based on Rough Sets. The ROSETTA System." Rough Sets in Knowledge discovery 1: Methodology and Applications **18 of Studies in Fuzziness and Soft Computing. Pulkowski and Skorn**(Cap 19): 376-399.

Olvera, J. A., J. Carrasco, et al. (2005). Sequential Search for Decremental Edition. 6th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2005, Brisbane, Australia, LNCS Springer-Verlag.

Olvera, J. A., J. F. Martínez, et al. (2005). Edition Schemes Based on BSE. CIARP 2005, M. Lazo and A. Sanfeliu (Eds.): LNCS. Springer-Verlag Berlin Heidelberg.

Orlowska, E., Ed. (1998). Incomplete Information, Rough sets analysis, Physica-Verlag.

Ovando, G., M. Bocco, et al. (2005). "Redes neuronales para modelar predicción de heladas." Agricultura Técnica **65**(1).

Pal, S. K. (2002). "Web mining in Soft Computing framework: Relevance, State of the art and Future Directions." IEEE Transactions on Neural Networks.

Pal, S. K. and P. Mitra (2001). Case-Generation: A Rough-Fuzzy Approach. International Conference on Case-Based Reasoning (ICCBR'01).

Pal, S. K. and P. Mitra (2003). Rough Sets, EM Algorithm, MST and Multispectral Image Segmentation. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China, LNCS26392003.

Pal, S. K. and A. Skowron (1999). Rough Fuzzy Hybridization: A New Trend in Decision-Making.

Parsons, S. (1996). "Current approaches to handling imperfect information in data and knowledges bases." IEEE Trans. On knowledge and data enginnering **8**(3).

Pawlak, Z. (1982). "Rough Sets." International journal of Computer and Information Sciences **11**: 341-356.

Pawlak, Z. (1995)a. "Rough Sets." Comm. of ACM. **38**(11).

Pawlak, Z. (1995)b. "Vagueness and uncertainty: a rough set perspective." Computational Intelligence. **11**(2).

Pelikan, M., D. Golberg, et al. (2002). Combining the Strengths of the Bayesian Optimization Algorithm and Adaptive Evolution Strategies. ILLIGAL Technical Reports & Publications series on Genetic Algorithms and Evolutionary Computation.

Peters, J. F. (2005). Rough Ethology: Towards a Biologically-Inspired Study of Collective Behavior in Intelligent Systems with Approximation Spaces. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Piñero, P. (2005). Un modelo para el aprendizaje y la clasificación automática basado en técnicas de softcomputing. Departamento de Ciencias de la Computación, Facultad de Matemática-Física y Computación. Universidad Central "Marta Abreu" de las Villas. Santa Clara, Cuba.

Piñero, P., L. Arco, et al. (2003). Two New Metrics for Feature Selection in Pattern Recognition. Lectures Notes in computer Science (LNCS 2905), Springer-Verlag 488-497.

Piñero, P., R. Grau, et al. (2002). Feature Selection in Statistical, Pattern Recognition and Machine Learning, Grupo de Inteligencia Artificial, Sherpa, Universidad Ciencias Informáticas.

Pyle, D., Ed. (1999). Data preparation for data mining, Morgan Kaufmann Publishers.

Quinlan, J. R. (1993). C-4.5: Programs for machine learning. San Mateo, California.

Quinlan, J. R. (2000). C5.0: An Informal Tutorial. Sidney.

Quintero, M., E. Amezquita, et al. (2003). "Guía para el uso de "Arboles de Decisión": Alternativas de Uso de la Tierra para los Llanos Orientales de Colombia. Estudio de Caso: Puerto López, Meta." Revista electrónica CIAT.

Ratke, C. and D. Andrade (2005). Using gain ratio distance (GR) to induce clustering. 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Wroclaw, Poland, IEEE Computer Society.

Revett, K., F. Gorunesco, et al. (2006). A Rough Sets based investigation of a Beta-Carotene/Retinol dataset. The International Symposium on Fuzzy and Rough Sets. ISFUROS2006, Santa Clara, Cuba.

Rich, E. and K. Knight, Eds. (1994). Artificial Intelligence. Madrid, España.

Riordan, D. and B. K. Hansen (2002). "A fuzzy case-based system for weather prediction." Engineering Intelligent Systems **3**: 139-146.

Ritter, G. L. W., H. B. Lowry, S. R. Isenhour, T. L. (1975). "An Algorithm for a Selective Nearest Neighbor Decision Rule." IEEE Transactions on Information Theory **216**: 665-669.

Rojas, J. E. (2006). Medición automática del D-score en imágenes de muestras de endometrio humano, Universidad Nacional de Colombia.

Romero, J., P. Machado, et al. (2006). Críticos de arte artificiales. Campus Multidisciplinar en Percepción e Inteligencia por los 50 Años de la Inteligencia Artificial, CMPI-2006, Albacete, España.

Rosemblatt, F. (1962). Principles of Neurodynamics. New York.

Rueda, R. E. (2006). "Modelos lluvia-escorrentía basados en redes neuronales artificiales." Ingeniería Civil: 127-136.

Ruiz, J., L. Aguila, et al. (1985). "Algoritmo BT y TB para el cálculo de todos los testores típicos." Revista Ciencias Matemáticas **VI**(2): 11-18.

Ruiz, R. (2006). Heurísticas de selección de atributos para datos de gran dimensionalidad. Departamento de Lenguajes y Sistemas Informáicos. Sevilla, Universidad de Sevilla.

Ruiz, R., J. Aguilar-Ruiz, et al. (2004). Wrapper for ranking feature selection. 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04), Springer Verlag.

Ruiz, R., J. Aguilar, et al. (2005). Analysis of feature rankings for classification. In Advances in Intelligent Data Analysis VI (IDA 2005), Springer Verlag.

Ruiz, R., J. Riquelme, et al. (2005). Heuristic search over a ranking for feature selection. Computational Intelligence and Bioinspired Systems. 8th International Work-Conference on Artificial Neural Networks (IWANN'05), Springer Verlag.

Saeys, Y., S. Degroeve, et al. (2003). "Fast feature selection using a simple Estimation of Distribution Algorithm : A case study on splice site prediction." Bioinformatics **1**(1): 1-10.

Sánchez, A. (2006). "Uso de árboles de decisiones." from http://www.monografias.com/trabajos41/arbol-de-decision/.

Sánchez, G., M. Lazo, et al. (1999). "Algoritmo genético para calcular testores típicos de costo mínimo." Memorias del IV Simposio Iberoamericano de Reconocimiento de Patrones SIARP'99, Ciudad Habana: 207-213.

Sánchez, J. S., R. Barandela, et al. (2003). "Analysis of new techniques to obtain quality training sets " Pattern Recognition Letters **24**(7): 1015-1022.

Santiesteban, Y. and A. Pons (2003). "LEX: un nuevo algoritmo para el cálculo de los testores típicos." Revista Ciencias Matemáticas **21**(1).

Santos, G. and A. José (2003). Algoritmo Genético para el cálculo de los -Testores Difusos. Instituto Nacional de Astrofísica Óptica y Electrónica.

Segovia, M. J. (2003). Predicción de insolvencias con el método Rough Set. España, Universidad Complutense de Madrid.

Simon, H. A. (1983). Why should machines learn? Machine Learning, An Artificial Intelligence approach. R. S. Michalski, J. G. Carbonell and T. M. Mitchel. Palo Alto, CA.

Skowron, A. (1999). New directions in Rough Sets, Data Mining, and Granular Soft Computing. 7th International Workshop (RSFDGRC'99), Yamaguchi, Japan, Lecture Notes in Artificial Intelligence 1711.

Skowron, A. and J. F. Peters (2003). Rough Sets: Trends and Challenges. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing 9th International Conference, RSFDGRC 2003, Chongqing, China, LNCS 26392003.

Skowron, A. and J. Stepaniuk (1992). Intelligent systems based on rough set approach. International Workshop Rough Sets. State of the Art and Perspectives.

Skowron, A., R. Swiniarski, et al. (2005). Approximation Spaces and Information Granulation. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Slezak, D. (2005). Rough Sets and Bayes Factor. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Slowinski, R. and D. Vanderpooten (1997). Similarity relation as a basis for rough approximations. Advances in Machine Intelligence & Soft-Computing. **IV:** 17-33.

Sordo, C. M. (2006). Técnicas Estadísticas para la Proyección Local de la Predicción Meteorológica Estacional. Métodos, Validación y Estudios de Sensibilidad. Dept. of Applied Mathematics and Computer Science. Santander, Spain, Universidad de Cantabria**:** 161.

Sugihara, K. and H. Tanaka (2006). Rough Sets approach to information systems with interval decision values in evaluation problems. The International Symposium on Fuzzy and Rough Sets. ISFUROS2006, Santa Clara, Cuba.

Suraj, Z. and P. Grochowalski (2005). The Rough Set Database System: An Overview. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Tay, F. E. and L. Shen (2003). "Fault diagnosis based on Rough Set Theory." Engineering Applications of Artificial Intelligence **16**: 39-43.

Tomek, I. (1976). "An Experiment with the edited nearest neighbor rule." IEEE Transactions on Systems, Man and Cybernetics **6**(6): 448-452.

Tsumoto, S. (2003). "Automated extraction of hierarchical decision rules from clinical databases using rough set model." Expert systems with Applications **24**: 189-197.

Vinterbo, S. V. (1999). Predictive Models in Medicine: Some Methods for Construction and Adaptation. Department of Computer and Information Science. Trondheim, Norwegian University of Science and Technology (NTNU).

Wang, G. Y. (2002). Attribute Core of Decision Table. Third International Conference, RSCTC 2002 Malvern, PA, USA, LNCS24752002 Rough Sets and Current Trends in Computing.

Wilson, D. L. (1972). "Asymptotic properties of Nearest Neighbor rules using edited data sets." IEEE Transactions on Systems, Man and Cybernetics, SMC **2**: 408-421.

Wilson, D. R. and T. R. Martínez (1997). "Improved Heterogeneous Distance Functions." Journal of Artificial Intelligence Research **6**: 1-34.

Wilson, R. and T. Martínez (1998). Reduction Techniques for ExemplarBased Learning Algorithms. Machine Learning, Computer Science Department, Brigham Young University. USA.

Wilson, R. and T. Martínez (2000). "Reduction Techniques for Instance Based Learning Algorithms." Machine Learning. Computer Science Department, Brigham Young University. USA **38**: 257-286.

Wilson, D. R. (1997). Advances in Instance-Based learning algorithms. Brigham, Department of Computer Science of Brigham Young University.

Witten, I. and E. Frank (2005)a. Implementations: Real machine learning schemes. Data Mining. Practical Machine Learning Tools and Techniques. I. Witten and E. Frank, Department of Computer Science. University of Waikato**:** 187-283.

Witten, I. and E. Frank (2005)b. Transformation: Engineering the input and output. Data Mining. Practical Machine Learning Tools and Techniques. I. Witten and E. Frank**:** 296-304.

Wolski, M. (2005). Formal Concept Analysis and Rough Set Theory from the Perspective of Finite Topological Approximations. Transactions on Rough Sets III. Lecture Notes in Computer Science. J. F. Peters and A. Skowron, Springer-Verlag GmbH. **3400**.

Wróblewski, J. (1995). Finding Minimal Reducts using Genetic Algorithm. Proc. of the Second Annual Join Conference on Information Sciences. Wrightsville Beach, NC. Also in: ICS Research report 16/95, Warsaw University of Technology.

Wróblewski, J. (2000). Ensembles of classifiers based on approximate reducts. Proc. of CSP 2000 Workshop, Informatik-Bericht Nr. 140. Humboldt-Universitat zu Berlin (2000). Revised and extended version in: Fundamenta Informaticae 47(3,4), IOS Press(2001).

Xu, M. and R. Setiono (2003). "Gene selection for cancer classification using a hybrid of univariate and multivariate feature selection methods." Applied Genomics and Proteomics **2**: 79-91.

Yao, Y. S. and C. Wong (1999). Methodologies for Knowledge Discovery and Data Mining. On Information-Theoretic Measures of attribute importance. Z. a. Zhou**:** 231-238.

Yao, Y. Y. (2003). On Generalizing Rough Set Theory. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China, LNCS 26392003.

Ye, D. and Z. Chen (2003). Inconsistency Classification and Discernibility-Matrix-Based Approaches for Computing an Attribute Core. Lecture Notes Artificial Intelligent (LNAI). C. Wang, Springer-Verlag**:** 269-273.

Yu, L. and H. Liu (2004)a. "Efficient feature selection via analysis of relevance and redundancy." Journal of machine learning research **5**: 1205-1224.

Yu, L. and H. Liu (2004)b. Redundancy based feature selection for microarry data. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Zhang, S., C. Zhang, et al. (2003). "Data preparation for data mining." Applied Artificial Intelligence **17**(5-6): 375-381.

Zhao, Y., H. Zhang, et al. (2003). Classification Using the Variable Precision Rough Set. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing 9th International Conference, RSFDGRC2003, Chongqing, China.

Zheng, Z., G. Wang, et al. (2003). A Rough Set and Rule Tree Based Incremental Knowledge Acquisition Algorithm. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. 9th International Conference, RSFDGRC2003, Chongqing, China.

Zhong, N., J. Dong, et al. (2001). "Using Rough sets with heuristics for feature selection." Journal of Intelligent Information Systems **16**: 199-214.

Zúñiga, A. and C. Jordán (2005). Aplicación de redes adaptables y sistemas de inferencia fuzzy para la previsión de caudales afluentes en centrales hidroeléctricas. Guayaquil, Ecuador, Facultad de Ingeniería en Electricidad y Computación. Escuela Politécnica del Litoral.

PRODUCCIÓN CIENTÍFICA DEL AUTOR SOBRE EL TEMA DE LA TESIS

Publicaciones en libros, revistas y memorias de eventos:

1. Piñero, P. y otros. “Two New Metrics for Feature Selection in Pattern Recognition”. Proceedings of 8th Iberoamerican Congress on Pattern Recognition, CIARP 2003. La Habana, mayo 2004. Book Series Lecture Notes in Computer Science, Publisher Springer Berlin/Heidelberg, ISSN 0302-9743, Volume 2905/2003, Book Progress in Pattern Recognition, Speech and Image Analysis, ISBN 978-3-540-20590-6, Pp. 488-497. 2003.

2. Caballero, Y. y otros. “Using rough sets to edit training sets in k-NN method”. Proceedings of Fifth International Conference on Intelligent Systems Design and Applications, ISDA’2005, Wroclaw, Polonia, 2005. IEEE Computer Society Order Number P2286. Library of Congress Number 2005930524. ISBN 0-7695-2286-6. 2005.

3. Caballero, Y. y otros. "Improving the k-NN method: Rough Set in edit training set”. Proceedings of The First IFIP International Conference on Artificial Intelligence in Theory and Practice (part of IFIP AI 2006). Santiago de Chile, agosto, 2006. Book Series IFIP International Federation for Information Processing, Publisher Springer Boston, ISSN 1571-5736 (Print) 1861-2288 (Online), Volume 218/2006, Book Professional Practice in Artificial Intelligence ISBN 978-0-387-34655-7, Pp. 21-30. 2006.

4. Caballero, Y. y otros. “Two new feature selection algorithms with Rough Sets Theory”. Proceedings of The First IFIP International Conference on Artificial Intelligence in Theory and Practice (part of IFIP AI 2006). Santiago de Chile, agosto, 2006. Book Series IFIP International Federation for Information Processing, Publisher Springer Boston, ISSN 1571-5736 (Print) 1861-2288 (Online), Volume 217/2006, Book Artificial Intelligence in Theory and Practice ISBN 978-0-387-34654-0, Pp. 209-216. 2006.

5. Caballero, Y. y otros. “A new measure based in the Rough Set Theory to estimate the training set quality”. Proceedings of 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC’06), Session Learning and

Fuzzy Control. Timisoara, Romania, septiembre, 2006. Publisher IEEE Computer Society, ISBN 0-7695-2740-X/06, Pp. 133-140.

6. Caballero, Y. y otros. “New Measures for Evaluating Decision Systems using Rough Set Theory: The Application in Seasonal Weather Forecasting”. Proceedings of Third International ICSC Symposium on Information Technologies in Environmental Engineering (ITEE'07). Carl von Ossietzky Universität Oldenburg. Alemania, marzo, 2007. Publisher Springer Verlag, ISBN 978-3-540-71334-0, Pp. 161-174.

7. Caballero, Y. y otros. “Rough Set Theory Measures for Evaluating Decision Systems”. Proceedings of 5th International Conference on Machine Learning and Data Mining in Pattern Recognition. Leipzing Alemania, julio, 2007. Publisher Springer Verlag in the LNAI Series in the book Machine Learning and Data Mining in Pattern Recognition, edited by Petra Porner.

8. Caballero, Y. y otros. “MRSReduct and RSRed*: Two new Feature Selection Algorithms using Rough Set Theory”. Proceedings of 4th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'07). Haikou, China, agosto, 2007. Publisher IEEE, indexed by both EI (Compendex) and ISTP.

9. Caballero, Y. y otros. “Knowledge Generation to Machine Learning using Rough Set Theory Measures”. Proceedings of 4th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'07). Haikou, China, agosto, 2007. Publisher IEEE, indexed by both EI (Compendex) and ISTP.

10. Caballero, Y. y otros. “Un nuevo algoritmo de selección de rasgos basado en la teoría de los conjuntos aproximados”. Proceedings of Campus Multidisciplinar en Percepción e Inteligencia por los 50 Años de la Inteligencia Artificial, CMPI-2006. Albacete España, julio 2006. Una perspectiva de la Inteligencia Artificial en su 50 Aniversario, ISBN 84-689-9561-4, Volumen II, Pp. 625-636.

11. Caballero, Y. y otros. “La Teoría de los Conjuntos Aproximados en la edición de conjuntos de entrenamiento para mejorar el desempeño del método k-NN”. Proceedings of Campus Multidisciplinar en Percepción e Inteligencia por los 50 Años de la Inteligencia Artificial, CMPI-2006. Albacete España, julio 2006. Una perspectiva de la

Inteligencia Artificial en su 50 Aniversario, ISBN 84-689-9561-4, Volumen II, Pp. 637-645.

12. Caballero, Y. y otros. “Un nuevo algoritmo de selección de rasgos basado en la teoría de los conjuntos aproximados”. Revista de Ingeniería de la Universidad de Antioquia. ISSN 0120-6230, Número 40, septiembre, 2007.
13. Caballero, Y. y otros. “A method to edit training set based on rough sets”. International Journal of Computational Intelligence Research (IJCIR), Research India Publications, India. ISSN 0973-1873, Volumen 3, Número 3, junio, 2007.
14. Caballero, Y. y otros. “El Reconocimiento de Patrones para la Relevancia de Rasgos”. Memorias de Tercer Evento Internacional de Matemática Educativa e Informática. Camagüey, noviembre 2002. ISBN 959-16-0179-4.
15. Caballero, Y. y otros. “Uso de los CA para el tratamiento de los datos”. Memorias de 8vo Congreso Nacional de Matemática y Computación COMPUMAT2003. Sancti Spíritus, noviembre, 2003. ISSN 1728 6042.
16. Caballero, Y. y otros. “Selección de rasgos relevantes para el análisis de datos”. Memorias de III Seminario Internacional de Matemática, Física e Informática Educativa. Camagüey, noviembre 2004. ISBN 959-16-0294-0.
17. Caballero, Y. R. y otros. “Selección de rasgos relevantes para el análisis de datos a través de los conjuntos aproximados” Memorias de I Simposio Iberoamericano de Inteligencia Artificial dentro de Informática’2004. La Habana, mayo 2004. ISBN 959-237-117-2.
18. Caballero, Y. y otros. “Estudio de conjuntos de entrenamientos para redes tipo MLP usando medidas de la teoría de los conjuntos aproximados”. Memorias de I Simposio Iberoamericano de Inteligencia Artificial dentro de Informática’2004. La Habana, mayo 2004. ISBN 959-237-117-2.
19. Caballero, Y. y otros. “Edición de conjuntos de entrenamiento usando los Conjuntos Aproximados”. Memorias de VIII Congreso de Nuevas Tecnologías y Aplicaciones

Informáticas dentro de Informática'2005. La Habana, mayo 2005. ISBN 959-7164-87-6.

20. Caballero, Y. y otros. "Una nueva medida para la estimación de la calidad de los conjuntos de entrenamiento usando la teoría de los conjuntos aproximados". Memorias de IV Congreso Nacional de Reconocimiento de Patrones. Sección Científica de la Sociedad Cubana de Matemática y Computación. La Habana, Cuba, octubre, 2006.
21. Caballero, Y. y otros. "Estimación de la calidad de los Conjuntos de Entrenamiento usando medidas de la Teoría de los Conjuntos Aproximados". Memorias de VII Conferencia Científica Internacional de la Unica. III Computer Sciences Workshop CSW'2006. Ciego de Ávila, Cuba, octubre, 2006. ISBN 959-16-0473-4.
22. Caballero, Y. y otros. "Rough Set Theory Measures for Quality Assessment of Training Set". Proceedings of International Symposium on Fuzzy and Rough Set ISFUROS'06. Santa Clara, diciembre, 2006. ISBN 959-250-307-9
23. Caballero, Y. y otros. "Nuevas medidas de la teoría de los conjuntos aproximados para la evaluación de sistemas de información en bioinformática". Memorias de II Congreso Internacional de Bioinformática y Neuroinformática. Informática 2007. La Habana, Cuba, febrero, 2007.
24. Caballero, Y. y otros. "La Teoría de los Conjuntos Aproximados en el mejoramiento de los conjuntos de entrenamiento en Bioinformática". Memorias de II Congreso Internacional de Bioinformática y Neuroinformática. Informática 2007. La Habana, Cuba, febrero, 2007.

Otros eventos aprobados:

(A estos eventos no se asistió por problemas con el financiamiento y la visa)

25. Caballero, Y. y otros. "A New Feature Selection Algorithm With Rough Sets Theory". The 2006 International Conference on Artificial Intelligence ICAI'06. Las Vegas, USA, junio, 2006.

26. Caballero, Y. y otros. “Una Propuesta para la Edición de Conjuntos de Entrenamiento Basada en la Teoría de los Conjuntos Aproximados”. 5ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2006. Orlando, Florida, EE.UU, julio, 2006.
27. Caballero, Y. y otros. “Rough Set in Edit Training Set to Improve MLP Neural Network and k-NN Method” World Multiconference on Systemics, Cybernetics and Informatics WMSCI 2006.Orlando, USA, julio, 2006.
28. Caballero, Y. y otros. “A New Method for Feature Selection by Using Rough Sets Theory”. World Multiconference on Systemics, Cybernetics and Informatics WMSCI 2006.Orlando, USA, julio, 2006.
29. Caballero, Y. y otros. “Rough Set theory measures for quality assessment of training set”. International Conference on Artificial Intelligence and Soft Computing ASC 2006. Palma De Mallorca, Spain, agosto, 2006.
30. Caballero, Y. y otros. “Edit training set through Rough Set Theory” 2006 International Conference on Intelligent Computing ICIC’06. Kunming, Yunnan Province, China, agosto, 2006.
31. Caballero, Y. y otros. “Efficiency assessment of training set through Rough Set measures”. International Conference on Intelligent Systems and Control ISC 2006. Honolulu, USA, agosto, 2006.
32. Caballero, Y. y otros. “Quality Assessment of Training Sets through Rough Set Measures”. IEEE International Conference on Information Reuse and Integration IEEE IRI 2006: Heuristic Systems Engineering. Hilton Waikoloa Village, Waikoloa, Hawaii, USA, septiembre, 2006.
33. Caballero, Y. y otros. “A new measure to quality assessment of the training set through Rough Set Theory”. IASTED International Conference on Artificial Intelligence and Applications, AIA 2007. Innsbruck, Austria, febrero, 2007.

Patentes y Registros de propiedad intelectual

- Registro de Software número 2331-2005 del Centro Nacional de Derecho de Autor a favor de: Sistema para la selección de rasgos relevantes (RoughSetsReduct). 2005.

- Registro de Software número 2529-2005 del Centro Nacional de Derecho de Autor a favor de: Sistema para la edición de conjuntos de entrenamiento (EditCE). 2005.
- Registro de Software número 3233-2006 del Centro Nacional de Derecho de Autor a favor de: Utilitario para el cálculo de las medidas de estimación de los Conjuntos Aproximados (MIRST 1.0). 2006.
- Registro de Software (*en proceso*) del Centro Nacional de Derecho de Autor a favor de: Sistema automatizado para el pronóstico de las temperaturas diarias en las estaciones de Camagüey (PROMETEM 1.0). 2007.

ANEXOS

Anexo 1. Conjuntos de datos del estudio experimental

Tabla A1.1 Descripción de los conjuntos de datos internacionales que se usaron en los experimentos

Conjuntos de Datos	Cantidad de instancias	Cantidad de atributos	Cantidad de Clases	Ausencia de información
Balance-Scale	625	4	3	no
Ballons	20	4	2	no
Breast-Cancer-Wisconsin	699	10	2	sí
Bupa (Liver Disorders)	345	6	2	no
Credit	876	20	2	sí
Credit-Screening	125	17	2	sí
Dermatology	366	34	6	sí
E-Coli	336	7	8	no
Exactly	780	13	2	sí
Glass	214	9	7	no
Hayes-Roth	133	4	3	sí
Heart-Disease (Hungarian)	294	13	5	sí
Hepatitis	155	19	2	sí
House-Votes	435	16	2	sí
Iris	150	4	3	no
Ionosphere	351	34	2	no
LED	226	24	10	no
Lung-Cancer	32	56	3	sí
Lymphography	148	18	4	no
M-of-N	1000	13	2	sí
Monks-1	432	7	2	no
Mushroom (Agaricus-Lepiota)	8124	22	2	sí
Page-blocks	5473	10	5	no
Pima-Indians-Diabetes	768	8	2	no
Postoperative patient	90	8	3	sí
Promoter-Gene-Sequence	106	57	2	no
Soybean (large)	307	35	19	sí
Tic-Tac-Toe	958	9	2	no
Waveform	5000	40	3	no
Wine Recognition	179	13	3	no
Yeast	1484	9	10	no
Zoo	101	17	7	no

Anexo 2. Algoritmo Generalized Editing

Pasos del Algoritmo Generalized Editing

Sea *X* Conjunto de Casos

P1. Para cada *x* en *X*:

p1.1. Encontrar el k-Vecino más Cercano de *x* en $X \setminus \{x\}$

p1.2. Si una clase tiene al menos *k'* representativos entre los k-Vecinos,

entonces se identifica *x* según esa clase (independiente de sus etiquetas de clases originales);

en caso contrario, descartar *x* de S_E

Anexo 3. Resultados experimentales de los métodos de selección de atributos

Tablas obtenidas por el SPSS relacionadas con el Experimento 1

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Longitud promedio UMDA+f3 - Longitud promedio UMDA+f1	Rangos negativos	60[a]	6.83	41.00
	Rangos positivos	50[b]	5.00	25.00
	Empates	120[c]		
	Total	230		
Longitud promedio Wroblewski(f1) - Longitud promedio UMDA+f1	Rangos negativos	10[d]	1.00	1.00
	Rangos positivos	210[e]	12.00	252.00
	Empates	10[f]		
	Total	230		
Longitud promedio Wroblewski(f2) - Longitud promedio UMDA+f1	Rangos negativos	0[g]	.00	.00
	Rangos positivos	210[h]	11.00	231.00
	Empates	20[i]		
	Total	230		
Longitud promedio Wroblewski(f3) - Longitud promedio UMDA+f1	Rangos negativos	10[j]	6.00	6.00
	Rangos positivos	180[k]	10.22	184.00
	Empates	40[l]		
	Total	230		

a. Longitud promedio UMDA+f3 < Longitud promedio UMDA+f1

b. Longitud promedio UMDA+f3 > Longitud promedio UMDA+f1

c. Longitud promedio UMDA+f1 = Longitud promedio UMDA+f3

d. Longitud promedio Wroblewski(f1) < Longitud promedio UMDA+f1

e. Longitud promedio Wroblewski(f1) > Longitud promedio UMDA+f1

f. Longitud promedio UMDA+f1 = Longitud promedio Wroblewski(f1)

g. Longitud promedio Wroblewski(f2) < Longitud promedio UMDA+f1

h. Longitud promedio Wroblewski(f2) > Longitud promedio UMDA+f1

i. Longitud promedio UMDA+f1 = Longitud promedio Wroblewski(f2)

j. Longitud promedio Wroblewski(f3) < Longitud promedio UMDA+f1

k. Longitud promedio Wroblewski(f3) > Longitud promedio UMDA+f1

l. Longitud promedio UMDA+f1 = Longitud promedio Wroblewski(f3)

Estadísticos de contraste[c,d]

			Longitud promedio UMDA+f3 - Longitud promedio UMDA+f1	Longitud promedio Wroblewski(f1) - Longitud promedio UMDA+f1	Longitud promedio Wroblewski(f2) - Longitud promedio UMDA+f1	Longitud promedio Wroblewski(f3) - Longitud promedio UMDA+f1
Z			-.712[a]	-4.078[b]	-4.026[b]	-3.589[b]
Sig. asintót. (bilateral)			.477	.000	.000	.000
Sig. Monte Carlo (bilateral)	Sig.		.500	.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.477	.000	.000	.000
		Límite superior	.522	.000	.000	.000
Sig. Monte Carlo (unilateral)	Sig.		.255	.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.243	.000	.000	.000
		Límite superior	.266	.000	.000	.000

a. Basado en los rangos positivos.

b. Basado en los rangos negativos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Longitud promedio Wroblewski(f1) - Longitud promedio UMDA+f3	Rangos negativos	10[a]	1.00	1.00
	Rangos positivos	210[b]	12.00	252.00
	Empates	10[c]		
	Total	230		
Longitud promedio Wroblewski(f2) - Longitud promedio UMDA+f3	Rangos negativos	0[d]	.00	.00
	Rangos positivos	210[e]	11.00	231.00
	Empates	20[f]		
	Total	230		
Longitud promedio Wroblewski(f3) - Longitud promedio UMDA+f3	Rangos negativos	20[g]	3.00	6.00
	Rangos positivos	160[h]	10.31	165.00
	Empates	50[i]		
	Total	230		

a. Longitud promedio Wroblewski(f1) < Longitud promedio UMDA+f3

b. Longitud promedio Wroblewski(f1) > Longitud promedio UMDA+f3

c. Longitud promedio UMDA+f3 = Longitud promedio Wroblewski(f1)

d. Longitud promedio Wroblewski(f2) < Longitud promedio UMDA+f3

e. Longitud promedio Wroblewski(f2) > Longitud promedio UMDA+f3

f. Longitud promedio UMDA+f3 = Longitud promedio Wroblewski(f2)

g. Longitud promedio Wroblewski(f3) < Longitud promedio UMDA+f3

h. Longitud promedio Wroblewski(f3) > Longitud promedio UMDA+f3

i. Longitud promedio UMDA+f3 = Longitud promedio Wroblewski(f3)

Estadísticos de contraste[b,c]

			Longitud promedio Wroblewski(f1) - Longitud promedio UMDA+f3	Longitud promedio Wroblewski(f2) - Longitud promedio UMDA+f3	Longitud promedio Wroblewski(f3) - Longitud promedio UMDA+f3
Z			-4.076[a]	-4.017[a]	-3.467[a]
Sig. asintót. (bilateral)			.000	.000	.001
Sig. Monte Carlo (bilateral)	Sig.		.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000
		Límite superior	.000	.000	.000
Sig. Monte Carlo (unilateral)	Sig.		.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000
		Límite superior	.000	.000	.000

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 624387341.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Tiempo promedio UMDA+f3 - Tiempo promedio UMDA+f1	Rangos negativos	10[a]	16.00	16.00
	Rangos positivos	220[b]	11.82	260.00
	Empates	0[c]		
	Total	230		
Tiempo promedio Wroblewski(f3) - Tiempo promedio UMDA+f1	Rangos negativos	10[d]	14.00	14.00
	Rangos positivos	220[e]	11.91	262.00
	Empates	0[f]		
	Total	230		
Tiempo promedio Wroblewski(f2) - Tiempo promedio UMDA+f1	Rangos negativos	10[g]	15.00	15.00
	Rangos positivos	220[h]	11.86	261.00
	Empates	0[i]		
	Total	230		
Tiempo promedio Wroblewski(f1) - Tiempo promedio UMDA+f1	Rangos negativos	10[j]	15.00	15.00
	Rangos positivos	220[k]	11.86	261.00
	Empates	0[l]		
	Total	230		

a. Tiempo promedio UMDA+f3 < Tiempo promedio UMDA+f1

b. Tiempo promedio UMDA+f3 > Tiempo promedio UMDA+f1

c. Tiempo promedio UMDA+f1 = Tiempo promedio UMDA+f3

d. Tiempo promedio Wroblewski(f3) < Tiempo promedio UMDA+f1

e. Tiempo promedio Wroblewski(f3) > Tiempo promedio UMDA+f1

f. Tiempo promedio UMDA+f1 = Tiempo promedio Wroblewski(f3)

g. Tiempo promedio Wroblewski(f2) < Tiempo promedio UMDA+f1

h. Tiempo promedio Wroblewski(f2) > Tiempo promedio UMDA+f1

i. Tiempo promedio UMDA+f1 = Tiempo promedio Wroblewski(f2)

j. Tiempo promedio Wroblewski(f1) < Tiempo promedio UMDA+f1

k. Tiempo promedio Wroblewski(f1) > Tiempo promedio UMDA+f1

l. Tiempo promedio UMDA+f1 = Tiempo promedio Wroblewski(f1)

Estadísticos de contraste[b,c]

			Tiempo promedio UMDA+f3 - Tiempo promedio UMDA+f1	Tiempo promedio Wroblewski(f3) - Tiempo promedio UMDA+f1	Tiempo promedio Wroblewski(f2) - Tiempo promedio UMDA+f1	Tiempo promedio Wroblewski(f1) - Tiempo promedio UMDA+f1
Z			-3.712[a]	-3.773[a]	-3.742[a]	-3.742[a]
Sig. asintót. (bilateral)			.000	.000	.000	.000
Sig. Monte Carlo (bilateral)	Sig.		.000	.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000	.000
		Límite superior	.000	.000	.000	.000
Sig. Monte Carlo (unilateral)	Sig.		.000	.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000	.000
		Límite superior	.000	.000	.000	.000

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Tiempo promedio Wroblewski(f3) - Tiempo promedio UMDA+f3	Rangos negativos	10[a]	23.00	23.00
	Rangos positivos	220[b]	11.50	253.00
	Empates	0[c]		
	Total	230		
Tiempo promedio Wroblewski(f2) - Tiempo promedio UMDA+f3	Rangos negativos	10[d]	23.00	23.00
	Rangos positivos	220[e]	11.50	253.00
	Empates	0[f]		
	Total	230		
Tiempo promedio Wroblewski(f1) - Tiempo promedio UMDA+f3	Rangos negativos	10[g]	22.00	22.00
	Rangos positivos	210[h]	11.00	231.00
	Empates	10[i]		
	Total	230		

a. Tiempo promedio Wroblewski(f3) < Tiempo promedio UMDA+f3

b. Tiempo promedio Wroblewski(f3) > Tiempo promedio UMDA+f3

c. Tiempo promedio UMDA+f3 = Tiempo promedio Wroblewski(f3)

d. Tiempo promedio Wroblewski(f2) < Tiempo promedio UMDA+f3

e. Tiempo promedio Wroblewski(f2) > Tiempo promedio UMDA+f3

f. Tiempo promedio UMDA+f3 = Tiempo promedio Wroblewski(f2)

g. Tiempo promedio Wroblewski(f1) < Tiempo promedio UMDA+f3

h. Tiempo promedio Wroblewski(f1) > Tiempo promedio UMDA+f3

i. Tiempo promedio UMDA+f3 = Tiempo promedio Wroblewski(f1)

Estadísticos de contraste[b,c]

			Tiempo promedio Wroblewski(f3) - Tiempo promedio UMDA+f3	Tiempo promedio Wroblewski(f2) - Tiempo promedio UMDA+f3	Tiempo promedio Wroblewski(f1) - Tiempo promedio UMDA+f3
Z			-3.499[a]	-3.499[a]	-3.394[a]
Sig. asintót. (bilateral)			.000	.000	.001
Sig. Monte Carlo (bilateral)	Sig.		.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000
		Límite superior	.001	.001	.000
Sig. Monte Carlo (unilateral)	Sig.		.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000
		Límite superior	.000	.000	.000

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 624387341.

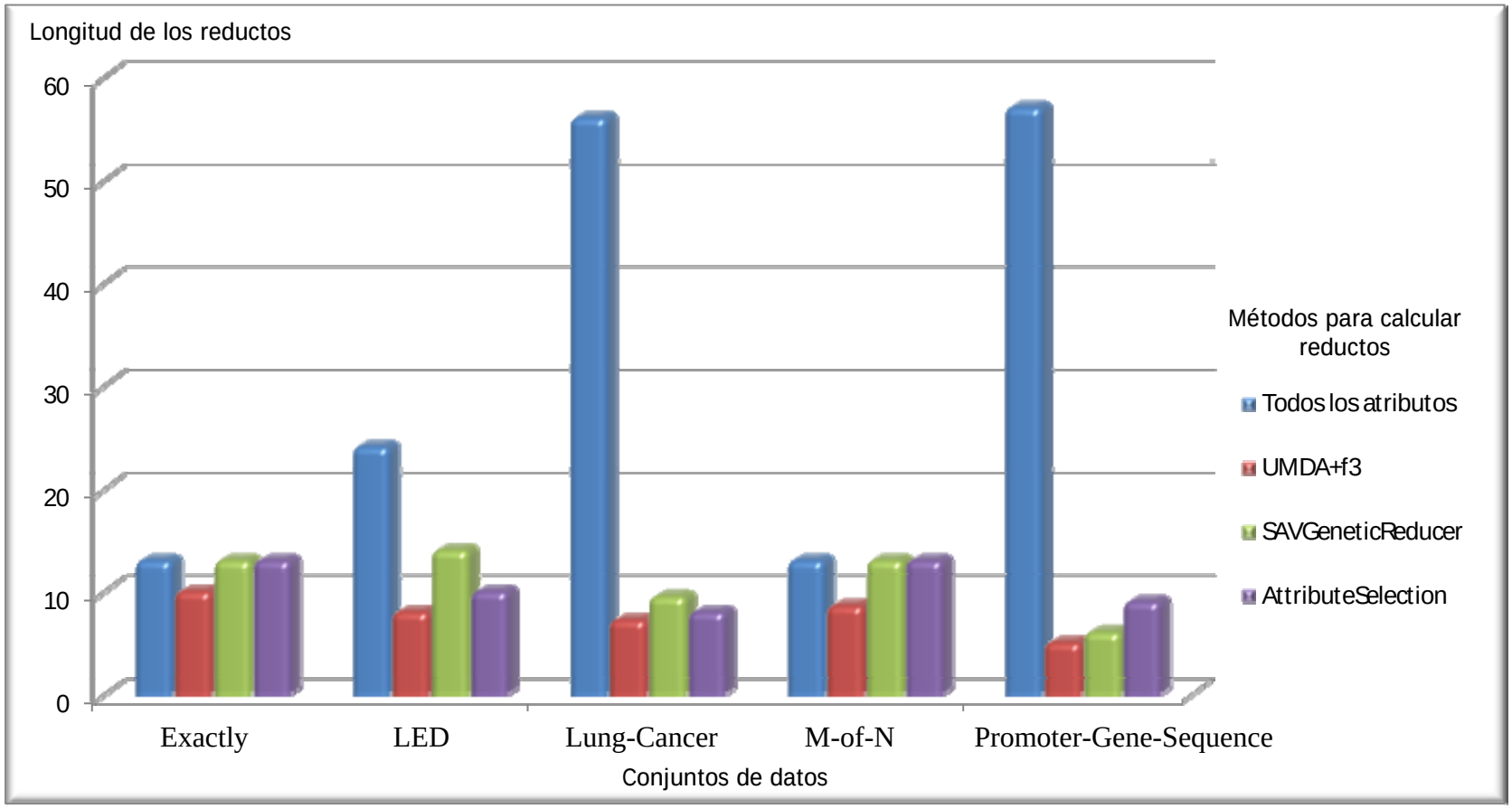

Figura A3.1 Longitud de los reductos obtenidos por diferentes métodos de cálculo de reductos

Tablas obtenidas por el SPSS relacionadas con el Experimento 2

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Longitud promedio UMDA+f1 - Cantidad de atributos	Rangos negativos	200[a]	10.00	190.00
	Rangos positivos	0[b]	.00	.00
	Empates	30[c]		
	Total	230		
Longitud promedio UMDA+f3 - Cantidad de atributos	Rangos negativos	210[d]	10.50	210.00
	Rangos positivos	0[e]	.00	.00
	Empates	20[f]		
	Total	230		

a. Longitud promedio UMDA+f1 < Cantidad de atributos

b. Longitud promedio UMDA+f1 > Cantidad de atributos

c. Cantidad de atributos = Longitud promedio UMDA+f1

d. Longitud promedio UMDA+f3 < Cantidad de atributos

e. Longitud promedio UMDA+f3 > Cantidad de atributos

f. Cantidad de atributos = Longitud promedio UMDA+f3

Estadísticos de contraste[b,c]

			Longitud promedio UMDA+f1 - Cantidad de atributos	Longitud promedio UMDA+f3 - Cantidad de atributos
Z			-3.829[a]	-3.921[a]
Sig. asintót. (bilateral)			.000	.000
Sig. Monte Carlo (bilateral)	Sig.		.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000
		Límite superior	.000	.000
Sig. Monte Carlo (unilateral)	Sig.		.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000
		Límite superior	.000	.000

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 299883525.

Prueba de Friedman

Rangos

	Rango promedio
Longitud promedio UMDA+f1	2.68
Longitud promedio UMDA+f3	2.89
Longitud promedio SAVGeneticReducer	2.45
Longitud promedio AttributeSelection	1.98

Estadísticos de contraste[a]

N			22
Chi-cuadrado			6.886
gl			3
Sig. asintót.			.076
Sig. Monte Carlo	Sig.		.075
	Intervalo de confianza de 95%	Límite inferior	.070
		Límite superior	.080

a. Prueba de Friedman

Tablas obtenidas por el SPSS relacionadas con el Experimento 3

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Longitud promedio RSRed*(H2) - Longitud promedio RSRed*(H1)	Rangos negativos	180[a]	9.50	171.00
	Rangos positivos	0[b]	.00	.00
	Empates	50[c]		
	Total	230		
Longitud promedio RSRed*(H1) - Longitud promedio RSReduct H1	Rangos negativos	130[d]	7.38	96.00
	Rangos positivos	20[e]	12.00	24.00
	Empates	80[f]		
	Total	230		
Longitud promedio RSRed*(H1) - Longitud promedio RSReduct H2	Rangos negativos	100[g]	5.50	55.00
	Rangos positivos	0[h]	.00	.00
	Empates	130[i]		
	Total	230		
Longitud promedio RSRed*(H1) - Longitud promedio RSReduct H3	Rangos negativos	130[j]	8.58	111.50
	Rangos positivos	30[k]	8.17	24.50
	Empates	70[l]		
	Total	230		
Longitud promedio EBR - Longitud promedio RSRed*(H1)	Rangos negativos	170[m]	9.59	163.00
	Rangos positivos	10[n]	8.00	8.00
	Empates	50[o]		
	Total	230		
Longitud promedio QReduct - Longitud promedio RSRed*(H1)	Rangos negativos	20[p]	5.00	10.00
	Rangos positivos	190[q]	11.11	200.00
	Empates	20[r]		
	Total	230		

a. Longitud promedio RSRed*(H2) < Longitud promedio RSRed*(H1)

b. Longitud promedio RSRed*(H2) > Longitud promedio RSRed*(H1)

c. Longitud promedio RSRed*(H1) = Longitud promedio RSRed*(H2)

d. Longitud promedio RSRed*(H1) < Longitud promedio RSReduct H1

e. Longitud promedio RSRed*(H1) > Longitud promedio RSReduct H1

f. Longitud promedio RSReduct H1 = Longitud promedio RSRed*(H1)

g. Longitud promedio RSRed*(H1) < Longitud promedio RSReduct H2

h. Longitud promedio RSRed*(H1) > Longitud promedio RSReduct H2

i. Longitud promedio RSReduct H2 = Longitud promedio RSRed*(H1)

j. Longitud promedio RSRed*(H1) < Longitud promedio RSReduct H3

k. Longitud promedio RSRed*(H1) > Longitud promedio RSReduct H3

l. Longitud promedio RSReduct H3 = Longitud promedio RSRed*(H1)

m. Longitud promedio EBR < Longitud promedio RSRed*(H1)

n. Longitud promedio EBR > Longitud promedio RSRed*(H1)

o. Longitud promedio RSRed*(H1) = Longitud promedio EBR

p. Longitud promedio QReduct < Longitud promedio RSRed*(H1)

q. Longitud promedio QReduct > Longitud promedio RSRed*(H1)

r. Longitud promedio RSRed*(H1) = Longitud promedio QReduct

Estadísticos de contraste[c,d]

			Longitud promedio RSRed*(H2) - Longitud promedio RSRed*(H1)	Longitud promedio RSRed*(H1) - Longitud promedio RSReduct H1	Longitud promedio RSRed*(H1) - Longitud promedio RSReduct H2	Longitud promedio RSRed*(H1) - Longitud promedio RSReduct H3	Longitud promedio EBR - Longitud promedio RSRed*(H1)	Longitud promedio QReduct - Longitud promedio RSRed*(H1)
Z			-3.758[a]	-2.089[a]	-3.051[a]	-2.253[a]	-3.398[a]	-3.553[b]
Sig. asintót. (bilateral)			.000	.037	.002	.024	.001	.000
Sig. Monte Carlo (bilateral)	Sig.		.000	.045	.002	.023	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.038	.001	.017	.000	.000
		Límite superior	.000	.053	.004	.028	.000	.000
Sig. Monte Carlo (unilateral)	Sig.		.000	.024	.001	.011	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.020	.000	.009	.000	.000
		Límite superior	.000	.028	.002	.014	.000	.000

a. Basado en los rangos positivos.

b. Basado en los rangos negativos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 1149983241.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Longitud promedio RSRed*(H2) - Longitud promedio RSReduct H1	Rangos negativos	210[a]	11.00	231.00
	Rangos positivos	0[b]	.00	.00
	Empates	20[c]		
	Total	230		
Longitud promedio RSRed*(H2) - Longitud promedio RSReduct H3	Rangos negativos	210[d]	11.00	231.00
	Rangos positivos	0[e]	.00	.00
	Empates	20[f]		
	Total	230		
Longitud promedio RSRed*(H2) - Longitud promedio RSReduct H2	Rangos negativos	190[g]	10.00	190.00
	Rangos positivos	0[h]	.00	.00
	Empates	40[i]		
	Total	230		
Longitud promedio EBR - Longitud promedio RSRed*(H2)	Rangos negativos	10[j]	1.00	1.00
	Rangos positivos	190[k]	11.00	209.00
	Empates	30[l]		
	Total	230		
Longitud promedio QReduct - Longitud promedio RSRed*(H2)	Rangos negativos	10[m]	2.50	2.50
	Rangos positivos	210[n]	11.43	228.50
	Empates	10[o]		
	Total	230		

a. Longitud promedio RSRed*(H2) < Longitud promedio RSReduct H1

b. Longitud promedio RSRed*(H2) > Longitud promedio RSReduct H1

c. Longitud promedio RSReduct H1 = Longitud promedio RSRed*(H2)

d. Longitud promedio RSRed*(H2) < Longitud promedio RSReduct H3

e. Longitud promedio RSRed*(H2) > Longitud promedio RSReduct H3

f. Longitud promedio RSReduct H3 = Longitud promedio RSRed*(H2)

g. Longitud promedio RSRed*(H2) < Longitud promedio RSReduct H2

h. Longitud promedio RSRed*(H2) > Longitud promedio RSReduct H2

i. Longitud promedio RSReduct H2 = Longitud promedio RSRed*(H2)

j. Longitud promedio EBR < Longitud promedio RSRed*(H2)

k. Longitud promedio EBR > Longitud promedio RSRed*(H2)

l. Longitud promedio RSRed*(H2) = Longitud promedio EBR

m. Longitud promedio QReduct < Longitud promedio RSRed*(H2)

n. Longitud promedio QReduct > Longitud promedio RSRed*(H2)

o. Longitud promedio RSRed*(H2) = Longitud promedio QReduct

Estadísticos de contraste[c,d]

			Longitud promedio RSRed*(H2) - Longitud promedio RSReduct H1	Longitud promedio RSRed*(H2) - Longitud promedio RSReduct H3	Longitud promedio RSRed*(H2) - Longitud promedio RSReduct H2	Longitud promedio EBR - Longitud promedio RSRed*(H2)	Longitud promedio QReduct - Longitud promedio RSRed*(H2)
Z			-4.045[a]	-4.018[a]	-3.841[a]	-3.893[b]	-3.928[b]
Sig. asintót. (bilateral)			.000	.000	.000	.000	.000
Sig. Monte Carlo (bilateral)	Sig.		.000	.000	.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000	.000	.000
		Límite superior	.000	.000	.000	.000	.000
Sig. Monte Carlo (unilateral)	Sig.		.000	.000	.000	.000	.000
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.000	.000	.000
		Límite superior	.000	.000	.000	.000	.000

a. Basado en los rangos positivos.

b. Basado en los rangos negativos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 205597102.

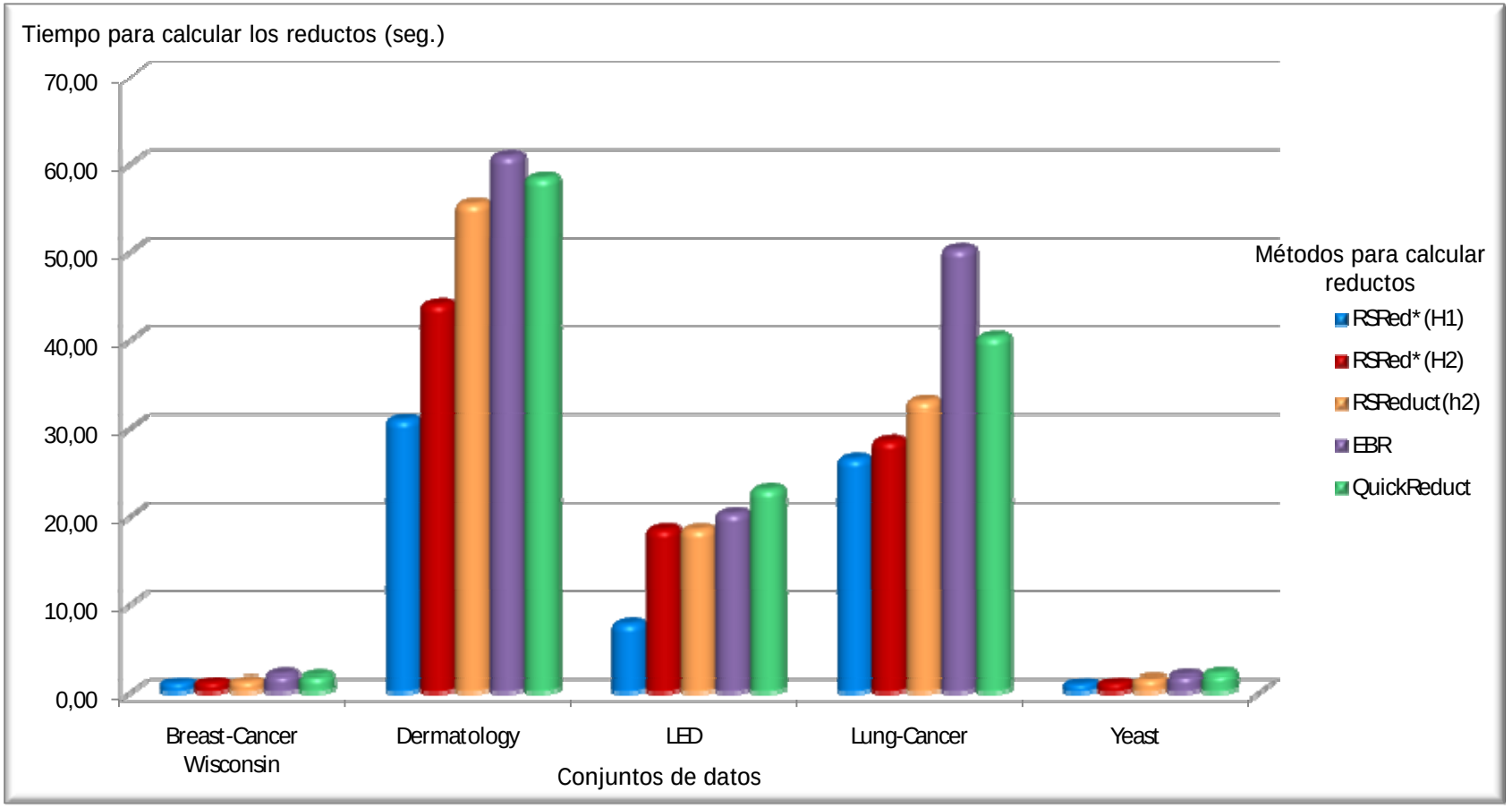

Figura A3.2 Tiempo empleado por diferentes métodos para calcular reductos

Tablas obtenidas por el SPSS relacionadas con el Experimento 4

Prueba de Friedman

Rangos

	Rango promedio
Accuracy promedio Todos Atributos	5.35
Accuracy promedio UMDA+f1	6.07
Accuracy promedio UMDA+f3	6.07
Accuracy promedio RSReduct H1	6.07
Accuracy promedio RSReduct H2	6.07
Accuracy promedio RSReduct H3	6.07
Accuracy promedio RSRed*(H2)	6.07
Accuracy promedio RSRed*(H1)	6.07
Accuracy promedio EBR	6.07
Accuracy promedio QReduct	6.07
Accuracy promedio SAVGeneticReducer	6.07

Estadísticos de contraste[a]

N			230
Chi-cuadrado			8.182
gl			10
Sig. asintót.			.611
Sig. Monte Carlo	Sig.		.732
	Intervalo de confianza de 95%	Límite inferior	.724
		Límite superior	.741

a. Prueba de Friedman

Tablas obtenidas por el SPSS relacionadas con el Experimento 5

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Longitud promedio RSRed*(H1) - Longitud promedio EBR	Rangos negativos	3[a]	5.67	17.00
	Rangos positivos	9[b]	6.78	61.00
	Empates	1[c]		
	Total	13		
Longitud promedio RSRed*(H1) - Longitud promedio RSAR	Rangos negativos	11[d]	6.82	75.00
	Rangos positivos	1[e]	3.00	3.00
	Empates	1[f]		
	Total	13		
Longitud promedio RSRed*(H2) - Longitud promedio EBR	Rangos negativos	8[g]	4.50	36.00
	Rangos positivos	0[h]	.00	.00
	Empates	5[i]		
	Total	13		
Longitud promedio RSRed*(H2) - Longitud promedio RSAR	Rangos negativos	12[j]	6.50	78.00
	Rangos positivos	0[k]	.00	.00
	Empates	1[l]		
	Total	13		
Longitud promedio RSRed*(H2) - Longitud promedio RSRed*(H1)	Rangos negativos	10[m]	5.50	55.00
	Rangos positivos	0[n]	.00	.00
	Empates	3[o]		
	Total	13		

a. Longitud promedio RSRed*(H1) < Longitud promedio EBR

b. Longitud promedio RSRed*(H1) > Longitud promedio EBR

c. Longitud promedio EBR = Longitud promedio RSRed*(H1)

d. Longitud promedio RSRed*(H1) < Longitud promedio RSAR

e. Longitud promedio RSRed*(H1) > Longitud promedio RSAR

f. Longitud promedio RSAR = Longitud promedio RSRed*(H1)

g. Longitud promedio RSRed*(H2) < Longitud promedio EBR

h. Longitud promedio RSRed*(H2) > Longitud promedio EBR

i. Longitud promedio EBR = Longitud promedio RSRed*(H2)

j. Longitud promedio RSRed*(H2) < Longitud promedio RSAR

k. Longitud promedio RSRed*(H2) > Longitud promedio RSAR

l. Longitud promedio RSAR = Longitud promedio RSRed*(H2)

m. Longitud promedio RSRed*(H2) < Longitud promedio RSRed*(H1)

n. Longitud promedio RSRed*(H2) > Longitud promedio RSRed*(H1)

o. Longitud promedio RSRed*(H1) = Longitud promedio RSRed*(H2)

Estadísticos de contraste[c,d]

			Longitud promedio RSRed*(H1) - Longitud promedio EBR	Longitud promedio RSRed*(H1) - Longitud promedio RSAR	Longitud promedio RSRed*(H2) - Longitud promedio EBR	Longitud promedio RSRed*(H2) - Longitud promedio RSAR	Longitud promedio RSRed*(H2) - Longitud promedio RSRed*(H1)
Z			-1.736[a]	-2.835[b]	-2.640[b]	-3.071[b]	-2.823[b]
Sig. asintót. (bilateral)			.083	.005	.008	.002	.005
Sig. Monte Carlo (bilateral)	Sig.		.088	.004	.008	.001	.002
	Intervalo de confianza de 99%	Límite inferior	.077	.001	.005	.000	.000
		Límite superior	.099	.006	.011	.002	.003
Sig. Monte Carlo (unilateral)	Sig.		.043	.002	.004	.000	.001
	Intervalo de confianza de 99%	Límite inferior	.038	.001	.002	.000	.000
		Límite superior	.048	.003	.005	.001	.001

a. Basado en los rangos negativos.

b. Basado en los rangos positivos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 112562564.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Longitud promedio UMDA+f1 - Longitud promedio AntRSAR	Rangos negativos	9[a]	5.00	45.00
	Rangos positivos	0[b]	.00	.00
	Empates	4[c]		
	Total	13		
Longitud promedio UMDA+f1 - Longitud promedio GenRSAR	Rangos negativos	13[d]	7.00	91.00
	Rangos positivos	0[e]	.00	.00
	Empates	0[f]		
	Total	13		
Longitud promedio UMDA+f3 - Longitud promedio AntRSAR	Rangos negativos	9[g]	7.67	69.00
	Rangos positivos	3[h]	3.00	9.00
	Empates	1[i]		
	Total	13		
Longitud promedio UMDA+f3 - Longitud promedio GenRSAR	Rangos negativos	13[j]	7.00	91.00
	Rangos positivos	0[k]	.00	.00
	Empates	0[l]		
	Total	13		

a. Longitud promedio UMDA+f1 < Longitud promedio AntRSAR

b. Longitud promedio UMDA+f1 > Longitud promedio AntRSAR

c. Longitud promedio AntRSAR = Longitud promedio UMDA+f1

d. Longitud promedio UMDA+f1 < Longitud promedio GenRSAR

e. Longitud promedio UMDA+f1 > Longitud promedio GenRSAR

f. Longitud promedio GenRSAR = Longitud promedio UMDA+f1

g. Longitud promedio UMDA+f3 < Longitud promedio AntRSAR

h. Longitud promedio UMDA+f3 > Longitud promedio AntRSAR

i. Longitud promedio AntRSAR = Longitud promedio UMDA+f3

j. Longitud promedio UMDA+f3 < Longitud promedio GenRSAR

k. Longitud promedio UMDA+f3 > Longitud promedio GenRSAR

l. Longitud promedio GenRSAR = Longitud promedio UMDA+f3

Estadísticos de contraste[b,c]

			Longitud promedio UMDA+f1 - Longitud promedio AntRSAR	Longitud promedio UMDA+f1 - Longitud promedio GenRSAR	Longitud promedio UMDA+f3 - Longitud promedio AntRSAR	Longitud promedio UMDA+f3 - Longitud promedio GenRSAR
Z			-2.724[a]	-3.180[a]	-2.354[a]	-3.238[a]
Sig. asintót. (bilateral)			.006	.001	.019	.001
Sig. Monte Carlo (bilateral)	Sig.		.005	.000	.017	.000
	Intervalo de confianza de 99%	Límite inferior	.002	.000	.012	.000
		Límite superior	.007	.001	.021	.000
Sig. Monte Carlo (unilateral)	Sig.		.003	.000	.009	.000
	Intervalo de confianza de 99%	Límite inferior	.001	.000	.006	.000
		Límite superior	.004	.001	.011	.000

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 329836257.

Comparación de los métodos de selección de atributos

Tabla A3.1 Comparación de los métodos teniendo en cuenta la longitud de los reductos construidos y el tiempo de ejecución

Nombre del algoritmo	Respecto a la Longitud de los reductos	Respecto al tiempo de ejecución
UMDA+f1 **UMDA+f3**	**Menor que los métodos:** AntRSAR GenRSAR Wróblewski(*f1*,*f2*,*f3*) **Similar a los métodos:** SavGeneticReducer AttributeSelection **Mayor que los métodos:** —	**Menor que los métodos:** Wróblewski(*f1*, *f2*, *f3*) UMDA+f1 menor que UMDA+f3 **Similar a los métodos:** — **Mayor que los métodos:** —
RSRed*(H1)	**Menor que los métodos:** RSReduct(h_1, h_3) QuickReduct RSAR **Similar a los métodos:** RSReduct(h_2) AttributeSelection **Mayor que los métodos:** EBR RSRed*(H2)	**Menor que los métodos:** RSReduct(h_1, h_2, h_3) EBR QuickReduct RSRed*(H2) **Similar a los métodos:** — **Mayor que los métodos:** —

RSRed*(H2)	**Menor que los métodos:** RSReduct(h_1, h_2, h_3) QuickReduct EBR RSRed*(H1) AttributeSelection RSAR **Similar a los métodos:** — **Mayor que los métodos:** —	**Menor que los métodos:** RSReduct(h_1, h_2, h_3) QuickReduct EBR **Similar a los métodos:** — **Mayor que los métodos:** RSRed*(H1)

Anexo 4. Resultados experimentales de los métodos de edición de conjuntos de entrenamiento

Tablas obtenidas por el SPSS relacionadas con el Experimento 6

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Precisión knn Edit3RS - Precisión knn Sin editar	Rangos negativos	0[a]	,00	,00
	Rangos positivos	150[b]	6,00	66,00
	Empates	0[c]		
	Total	150		

a. Precisión knn Edit3RS < Precisión knn Sin editar

b. Precisión knn Edit3RS > Precisión knn Sin editar

c. Precisión knn Sin editar = Precisión knn Edit3RS

Estadísticos de contraste[b,c]

			Precisión knn Edit3RS - Precisión knn Sin editar
Z			-2,934[a]
Sig. asintót. (bilateral)			,003
Sig. Monte Carlo (bilateral)	Sig.		,001
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,002
Sig. Monte Carlo (unilateral)	Sig.		,001
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,001

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 299883525.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Precisión knn ENN - Precisión knn Edit3RS	Rangos negativos	110[a]	6,00	54,00
	Rangos positivos	20[b]	1,00	1,00
	Empates	20[c]		
	Total	150		

a. Precisión knn ENN < Precisión knn Edit3RS

b. Precisión knn ENN > Precisión knn Edit3RS

c. Precisión knn Edit3RS = Precisión knn ENN

Estadísticos de contraste[b,c]

			Precisión knn ENN - Precisión knn Edit3RS
Z			-2,701[a]
Sig. asintót. (bilateral)			,007
Sig. Monte Carlo (bilateral)	Sig.		,005
	Intervalo de confianza de 99%	Límite inferior	,002
		Límite superior	,007
Sig. Monte Carlo (unilateral)	Sig.		,002
	Intervalo de confianza de 99%	Límite inferior	,001
		Límite superior	,003

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 334431365.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Precisión knn Todoknn - Precisión knn Edit3RS	Rangos negativos	100[a]	6,17	37,00
	Rangos positivos	30[b]	2,67	8,00
	Empates	20[c]		
	Total	150		

a. Precisión knn Todoknn < Precisión knn Edit3RS

b. Precisión knn Todoknn > Precisión knn Edit3RS

c. Precisión knn Edit3RS = Precisión knn Todoknn

Estadísticos de contraste[b,c]

			Precisión knn Todoknn - Precisión knn Edit3RS
Z			-1,718[a]
Sig. asintót. (bilateral)			,086
Sig. Monte Carlo (bilateral)	Sig.		,094
	Intervalo de confianza de 99%	Límite inferior	,083
		Límite superior	,105
Sig. Monte Carlo (unilateral)	Sig.		,049
	Intervalo de confianza de 99%	Límite inferior	,044
		Límite superior	,055

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 743671174.

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Precisión knn Edicion Generalizada - Precisión knn Edit3RS	Rangos negativos	140[a]	6,50	65,00
	Rangos positivos	10[b]	1,00	1,00
	Empates	0[c]		
	Total	150		

a. Precisión knn Edicion Generalizada < Precisión knn Edit3RS

b. Precisión knn Edicion Generalizada > Precisión knn Edit3RS

c. Precisión knn Edit3RS = Precisión knn Edicion Generalizada

Estadísticos de contraste[b,c]

			Precisión knn Edicion Generalizada - Precisión knn Edit3RS
Z			-2,845[a]
Sig. asintót. (bilateral)			,004
Sig. Monte Carlo (bilateral)	Sig.		,002
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,003
Sig. Monte Carlo (unilateral)	Sig.		,001
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,002

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 221623949.

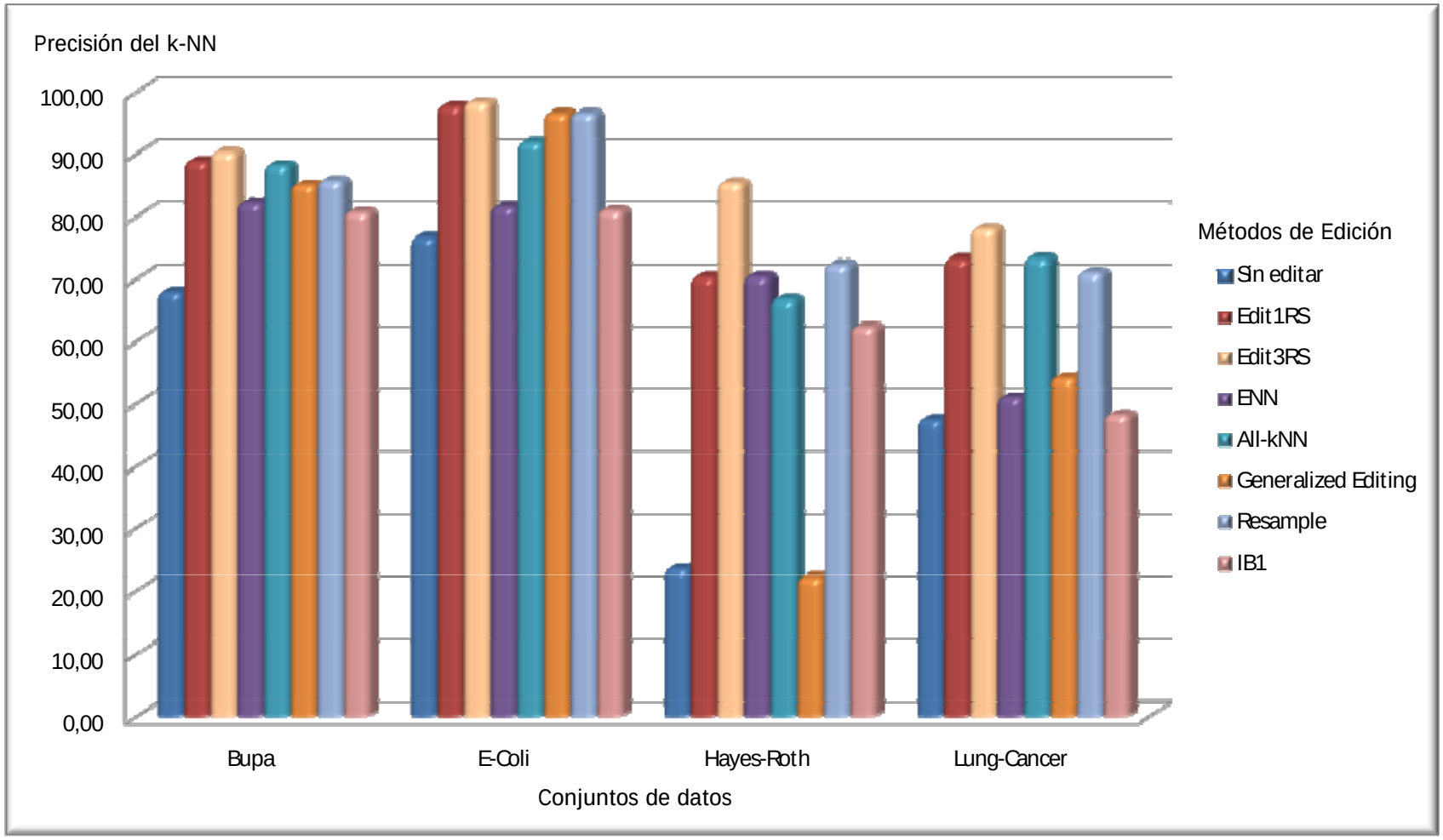

Figura A4.1 Precisión del clasificador k-Vecinos más Cercanos con muestras editadas por diferentes métodos

Tablas obtenidas por el SPSS relacionadas con el Experimento 7

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Retención Edit3RS - Retención Edit1RS	Rangos negativos	10[a]	1.00	1.00
	Rangos positivos	140[b]	6.50	65.00
	Empates	0[c]		
	Total	150		
Retención ENN - Retención Edit1RS	Rangos negativos	10[d]	8.00	8.00
	Rangos positivos	140[e]	5.80	58.00
	Empates	0[f]		
	Total	150		
RetenciónTodoknn - Retención Edit1RS	Rangos negativos	20[g]	6.50	13.00
	Rangos positivos	130[h]	5.89	53.00
	Empates	0[i]		
	Total	150		
Retención Edicion Generalizada - Retención Edit1RS	Rangos negativos	10[j]	7.00	7.00
	Rangos positivos	140[k]	5.90	59.00
	Empates	0[l]		
	Total	150		

a. Retención Edit3RS < Retención Edit1RS

b. Retención Edit3RS > Retención Edit1RS

c. Retención Edit1RS = Retención Edit3RS

d. Retención ENN < Retención Edit1RS

e. Retención ENN > Retención Edit1RS

f. Retención Edit1RS = Retención ENN

g. RetenciónTodoknn < Retención Edit1RS

h. RetenciónTodoknn > Retención Edit1RS

i. Retención Edit1RS = RetenciónTodoknn

j. Retención Edicion Generalizada < Retención Edit1RS

k. Retención Edicion Generalizada > Retención Edit1RS

l. Retención Edit1RS = Retención Edicion Generalizada

Estadísticos de contraste[b,c]

			Retención Edit3RS - Retención Edit1RS	Retención ENN - Retención Edit1RS	Retención Todoknn - Retención Edit1RS	Retención Edicion Generalizada - Retención Edit1RS
Z			-2.845[a]	-2.223[a]	-1.778[a]	-2.312[a]
Sig. asintót. (bilateral)			.004	.026	.075	.021
Sig. Monte Carlo (bilateral)	Sig.		.002	.023	.085	.018
	Intervalo de confianza de 99%	Límite inferior	.000	.018	.074	.013
		Límite superior	.003	.029	.095	.023
Sig. Monte Carlo (unilateral)	Sig.		.001	.013	.042	.010
	Intervalo de confianza de 99%	Límite inferior	.000	.010	.037	.008
		Límite superior	.002	.016	.047	.013

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

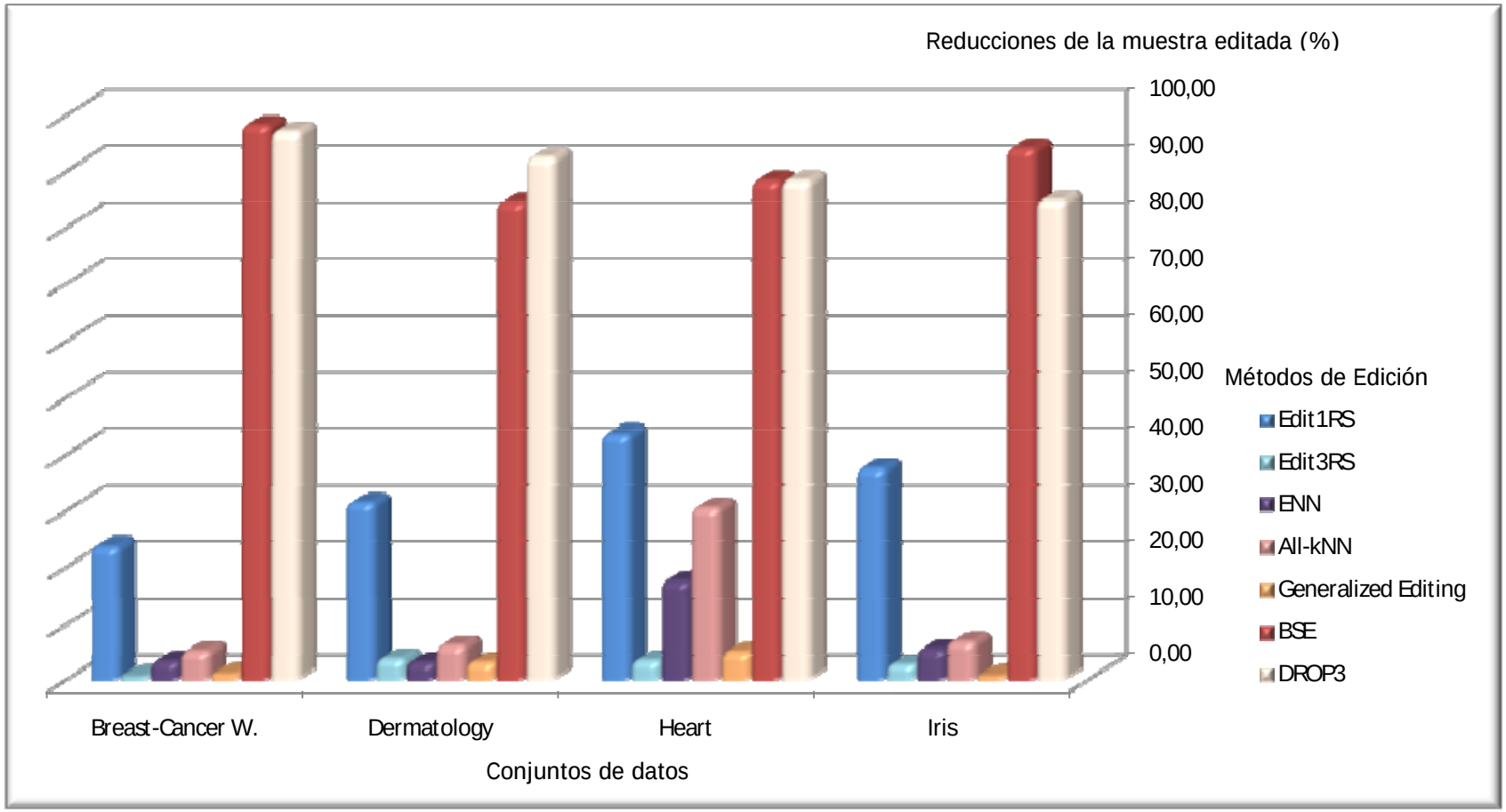

Figura A4.2 Reducciones de muestras editadas por diferentes métodos

Tablas obtenidas por el SPSS relacionadas con el Experimento 8

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Tiempo Edit3RS - Tiempo Edit1RS	Rangos negativos	0[a]	.00	.00
	Rangos positivos	140[b]	7.50	105.00
	Empates	10[c]		
	Total	150		
tiempo All kNN - Tiempo Edit1RS	Rangos negativos	5[d]	3.00	3.00
	Rangos positivos	145[e]	8.87	133.00
	Empates	0[f]		
	Total	150		
Tiempo Edición Generalizada - Tiempo Edit1RS	Rangos negativos	20[g]	7.80	39.00
	Rangos positivos	130[h]	8.82	97.00
	Empates	0[i]		
	Total	150		
Teimpo Multiedit - Tiempo Edit1RS	Rangos negativos	10[j]	6.00	30.00
	Rangos positivos	140[k]	9.00	90.00
	Empates	0[l]		
	Total	150		
tiempo All kNN - Tiempo Edit3RS	Rangos negativos	4[m]	10.00	20.00
	Rangos positivos	146[n]	8.29	116.00
	Empates	0[o]		
	Total	150		

a. Tiempo Edit3RS < Tiempo Edit1RS

b. Tiempo Edit3RS > Tiempo Edit1RS

c. Tiempo Edit1RS = Tiempo Edit3RS

d. tiempo All kNN < Tiempo Edit1RS

e. tiempo All kNN > Tiempo Edit1RS

f. Tiempo Edit1RS = tiempo All kNN

g. Tiempo Edición Generalizada < Tiempo Edit1RS

h. Tiempo Edición Generalizada > Tiempo Edit1RS

i. Tiempo Edit1RS = Tiempo Edición Generalizada

j. Teimpo Multiedit < Tiempo Edit1RS

k. Teimpo Multiedit > Tiempo Edit1RS

l. Tiempo Edit1RS = Teimpo Multiedit

m. tiempo All kNN < Tiempo Edit3RS

n. tiempo All kNN > Tiempo Edit3RS

o. Tiempo Edit3RS = tiempo All kNN

Estadísticos de contraste[b,c]

			Tiempo Edit3RS - Tiempo Edit1RS	tiempo All kNN - Tiempo Edit1RS	Tiempo Edición Generalizada - Tiempo Edit1RS	Teimpo Multiedit - Tiempo Edit1RS	tiempo All kNN - Tiempo Edit3RS
Z			-3.296[a]	-3.361[a]	-1.500[a]	-1.704[a]	-2.482[a]
Sig. asintót. (bilateral)			.001	.001	.134	.088	.013
Sig. Monte Carlo (bilateral)	Sig.		.000	.000	.144	.091	.011
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.130	.080	.007
		Límite superior	.000	.000	.157	.102	.015
Sig. Monte Carlo (unilateral)	Sig.		.000	.000	.071	.043	.006
	Intervalo de confianza de 99%	Límite inferior	.000	.000	.064	.038	.004
		Límite superior	.000	.000	.077	.048	.008

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

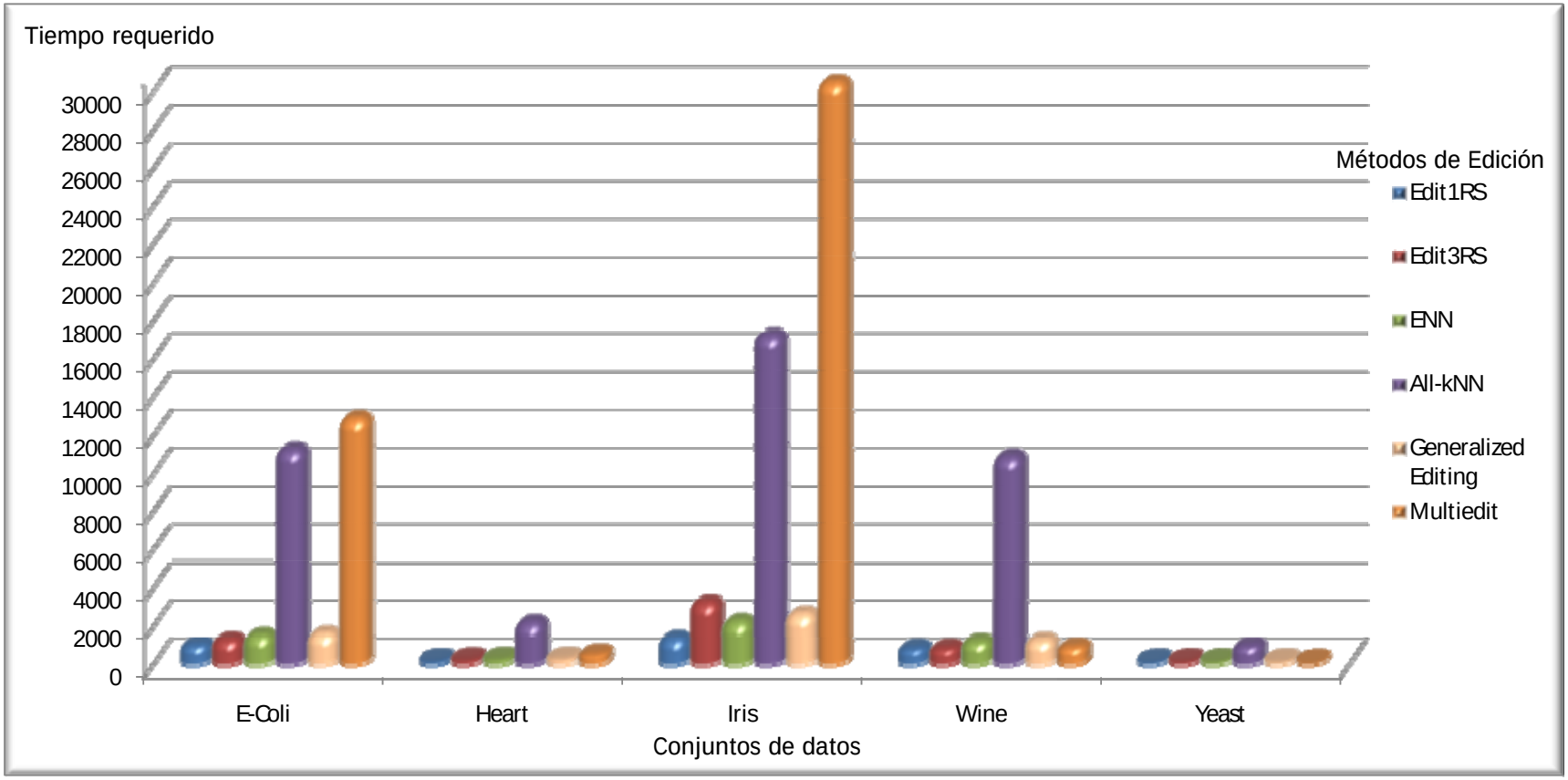

Figura A4.3 Tiempo requerido por diferentes métodos para editar conjuntos de entrenamiento

Tablas obtenidas por el SPSS relacionadas con el Experimento 9

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Precisión Resample - Precisión Edit3RS	Rangos negativos	58[a]	65,03	3771,50
	Rangos positivos	83[b]	75,17	6239,50
	Empates	9[c]		
	Total	150		

a. Precisión Resample < Precisión Edit3RS

b. Precisión Resample > Precisión Edit3RS

c. Precisión Edit3RS = Precisión Resample

Estadísticos de contraste[b,c]

			Precisión Resample - Precisión Edit3RS
Z			-2,540[a]
Sig. asintót. (bilateral)			,011
Sig. Monte Carlo (bilateral)	Sig.		,011
	Intervalo de confianza de 99%	Límite inferior	,007
		Límite superior	,014
Sig. Monte Carlo (unilateral)	Sig.		,006
	Intervalo de confianza de 99%	Límite inferior	,004
		Límite superior	,008

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

Tablas obtenidas por el SPSS relacionadas con el Experimento 10

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Precisión knn IB1 - Precisión knn Edit1RS	Rangos negativos	145[a]	8.23	107.00
	Rangos positivos	5[b]	6.50	13.00
	Empates	0[c]		
	Total	150		
Precisión knn IB1 - Precisión knn Edit3RS	Rangos negativos	148[d]	9.17	110.00
	Rangos positivos	2[e]	3.33	10.00
	Empates	0[f]		
	Total	150		

a. Precisión knn IB1 < Precisión knn Edit1RS

b. Precisión knn IB1 > Precisión knn Edit1RS

c. Precisión knn Edit1RS = Precisión knn IB1

d. Precisión knn IB1 < Precisión knn Edit3RS

e. Precisión knn IB1 > Precisión knn Edit3RS

f. Precisión knn Edit3RS = Precisión knn IB1

Estadísticos de contraste[b,c]

			Precisión knn IB1 - Precisión knn Edit1RS	Precisión knn IB1 - Precisión knn Edit3RS
Z			-2.669[a]	-2.840[a]
Sig. asintót. (bilateral)			.008	.005
Sig. Monte Carlo (bilateral)	Sig.		.005	.003
	Intervalo de confianza de 99%	Límite inferior	.002	.001
		Límite superior	.007	.005
Sig. Monte Carlo (unilateral)	Sig.		.002	.001
	Intervalo de confianza de 99%	Límite inferior	.001	.000
		Límite superior	.004	.002

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

Tablas obtenidas por el SPSS relacionadas con el Experimento 11

Prueba de los rangos con signo de Wilcoxon

Rangos

		N	Rango promedio	Suma de rangos
Acc knn ENN+BSE - Acc knn Edit3RS	Rangos negativos	8[a]	5.75	46.00
	Rangos positivos	2[b]	4.50	9.00
	Empates	0[c]		
	Total	10		
Acc knn DROP3+BSE - Acc knn Edit3RS	Rangos negativos	9[d]	5.78	52.00
	Rangos positivos	1[e]	3.00	3.00
	Empates	0[f]		
	Total	10		
Acc knn DROP4+BSE - Acc knn Edit3RS	Rangos negativos	9[g]	5.56	50.00
	Rangos positivos	1[h]	5.00	5.00
	Empates	0[i]		
	Total	10		
Acc knn ICF - Acc knn Edit3RS	Rangos negativos	10[j]	5.50	55.00
	Rangos positivos	0[k]	.00	.00
	Empates	0[l]		
	Total	10		
Acc knn DROP3 - Acc knn Edit3RS	Rangos negativos	10[m]	5.50	55.00
	Rangos positivos	0[n]	.00	.00
	Empates	0[o]		
	Total	10		

a. Acc knn ENN+BSE < Acc knn Edit2RS

b. Acc knn ENN+BSE > Acc knn Edit2RS

c. Acc knn Edit3RS = Acc knn ENN+BSE

d. Acc knn DROP3+BSE < Acc knn Edit3RS

e. Acc knn DROP3+BSE > Acc knn Edit3RS

f. Acc knn Edit3RS = Acc knn DROP3+BSE

g. Acc knn DROP4+BSE < Acc knn Edit3RS

h. Acc knn DROP4+BSE > Acc knn Edit3RS

i. Acc knn Edit3RS = Acc knn DROP4+BSE

j. Acc knn ICF < Acc knn Edit3RS

k. Acc knn ICF > Acc knn Edit3RS

l. Acc knn Edit3RS = Acc knn ICF

m. Acc knn DROP3 < Acc knn Edit3RS

n. Acc knn DROP3 > Acc knn Edit3RS

o. Acc knn Edit3RS = Acc knn DROP3

Estadísticos de contraste[b,c]

			Acc knn ENN+BSE - Acc knn Edit3RS	Acc knn DROP3+BSE - Acc knn Edit3RS	Acc knn DROP4+BSE - Acc knn Edit3RS	Acc knn ICF - Acc knn Edit3RS	Acc knn DROP3 - Acc knn Edit3RS
Z			-1.886[a]	-2.497[a]	-2.293[a]	-2.803[a]	-2.803[a]
Sig. asintót. (bilateral)			.059	.013	.022	.005	.005
Sig. Monte Carlo (bilateral)	Sig.		.064	.010	.019	.002	.002
	Intervalo de confianza de 99%	Límite inferior	.055	.006	.014	.000	.000
		Límite superior	.073	.013	.024	.004	.004
Sig. Monte Carlo (unilateral)	Sig.		.029	.004	.008	.001	.001
	Intervalo de confianza de 99%	Límite inferior	.025	.002	.006	.000	.000
		Límite superior	.034	.005	.010	.001	.001

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

Comparación de los métodos de edición de conjuntos de entrenamiento

Tabla A4.1 Comparación de los métodos teniendo en cuenta la precisión de la clasificación, el tiempo de ejecución y las reducciones de la muestra original

Nombre del algoritmo	Error cuadrático medio de la clasificación	Precisión de la clasificación	Tiempo de ejecución	Reducciones
Edit1RS	**Menor que los métodos:** ENN All-kNN GE Multiedit Resample IB1 IB2	**Superior a los métodos:** ENN All-kNN GE IB1 ICF DROP3	**Menor que los métodos:** Edit3RS All-kNN GE Multiedit	**Mayor que los métodos:** Edit3RS ENN All-kNN GE
	Similar a los métodos: __	**Similar a los métodos:** IB2 Multiedit ENN+BSE DROP3+BSE DROP4+BSE DROP5+BSE	**Similar a los métodos:** ENN	**Similar a los métodos:** Multiedit ICF
	Mayor que los métodos: Edit3RS	**Peor que los métodos:** Edit3RS Resample BSE DROP4	**Mayor que los métodos:** __	**Menor que los métodos:** BSE DROP3 DROP4 ENN+BSE DROP3+BSE DROP4+BSE DROP5+BSE

Edit3RS	**Menor que los métodos:** Edit1RS ENN All-kNN GE Multiedit Resample IB1 IB2	**Superior a los métodos:** Edit1RS ENN All-kNN GE IB1 Resample ICF DROP3 DROP4 ENN+BSE DROP3+BSE DROP4+BSE DROP5+BSE	**Menor que los métodos:** All-kNN	**Mayor que los métodos:** —
	Similar a los métodos: —	**Similar a los métodos:** IB2 Multiedit BSE	**Similar a los métodos:** GE Multiedit	**Similar a los métodos:** GE
	Mayor que los métodos: —	**Peor que los métodos:** —	**Mayor que los métodos:** Edit1RS ENN	**Menor que los métodos:** Edit1RS ENN All-kNN Multiedit BSE DROP3 DROP4 ENN+BSE DROP3+BSE DROP4+BSE DROP5+BSE

Anexo 5. Correlaciones estadísticas entre las medidas de inferencia y las precisiones de los clasificadores (desempeño general)

Tabla A5.1 Correlaciones entre las medidas de inferencia y la precisión del MLP

		Calidad de la Calsificación	Precisión generalizada de la clasificación	Calidad generalizada de la clasificación	Coeficiente de aproximación general
Precisión MLP	**Correlación de Pearson**	.679	.743	.678	.767
	Sig. (bilateral)	.000	.000	.000	.000
	N	250	250	250	250

Tabla A5.2 Correlaciones entre las medidas de inferencia y la precisión del k-NN

		Calidad de la Calsificación	Precisión generalizada de la clasificación	Calidad generalizada de la clasificación	Coeficiente de aproximación general
Precisión kNN	**Correlación de Pearson**	.715	.773	.740	.746
	Sig. (bilateral)	.000	.000	.000	.000
	N	250	250	250	250

Tabla A5.3 Correlaciones entre las medidas de inferencia y la precisión del C4.5

		Calidad de la Calsificación	Precisión generalizada de la clasificación	Calidad generalizada de la clasificación	Coeficiente de aproximación general
Precisión C4.5	**Correlación de Pearson**	.718	.791	.742	.794
	Sig. (bilateral)	.000	.000	.000	.000
	N	250	250	250	250

Anexo 6. Correlaciones estadísticas entre las medidas de inferencia y las precisiones de los clasificadores (desempeño por clases)

Tabla A6.1 Correlaciones entre las medidas de inferencia y la precisión por clases del clasificador MLP

		Precisión de la aproximación	Calidad de la aproximación
Precisión por clase MLP	**Correlación de Pearson**	.135	.102
	Sig. (bilateral)	.001	.015
	N	570	570

Tabla A6.2 Correlaciones entre las medidas de inferencia y la precisión por clases del clasificador k-NN

		Precisión de la aproximación	Calidad de la aproximación
Precisión por clase kNN	**Correlación de Pearson**	.232	.209
	Sig. (bilateral)	.000	.000
	N	570	570

Tabla A6.3 Correlaciones entre las medidas de inferencia y la precisión por clases del clasificador C4.5

		Precisión de la aproximación	Calidad de la aproximación
Precisión por clase C45	**Correlación de Pearson**	.205	.177
	Sig. (bilateral)	.000	.000
	N	570	570

Anexo 7. Generación del conocimiento a partir de las medidas de inferencia y las precisiones de los clasificadores

Tabla A7.1 Percentiles calculados para la precisión de los clasificadores

Estadísticos

Precisión C45

N	Válidos	250
	Perdidos	0
Percentiles	20	72.72730
	40	78.76070
	60	88.42574
	80	94.87180

Estadísticos

Precisión kNN

N	Válidos	250
	Perdidos	0
Percentiles	20	70.29032
	40	80.71676
	60	86.84876
	80	95.45450

Estadísticos

Precisión MLP

N	Válidos	250
	Perdidos	0
Percentiles	20	74.66664
	40	83.72090
	60	90.91164
	80	96.27684

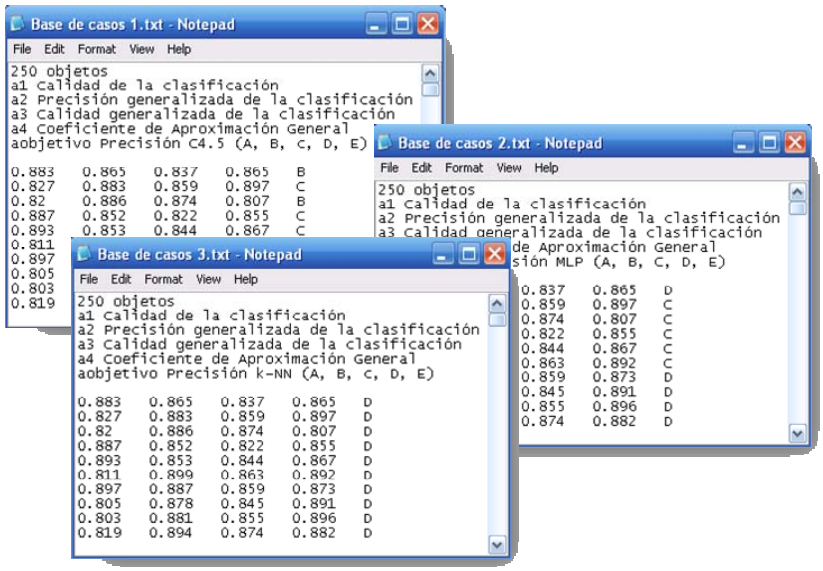

Figura A7.1 Conjuntos de datos construidos con los resultados de las medidas y la precisión discretizada de los clasificadores

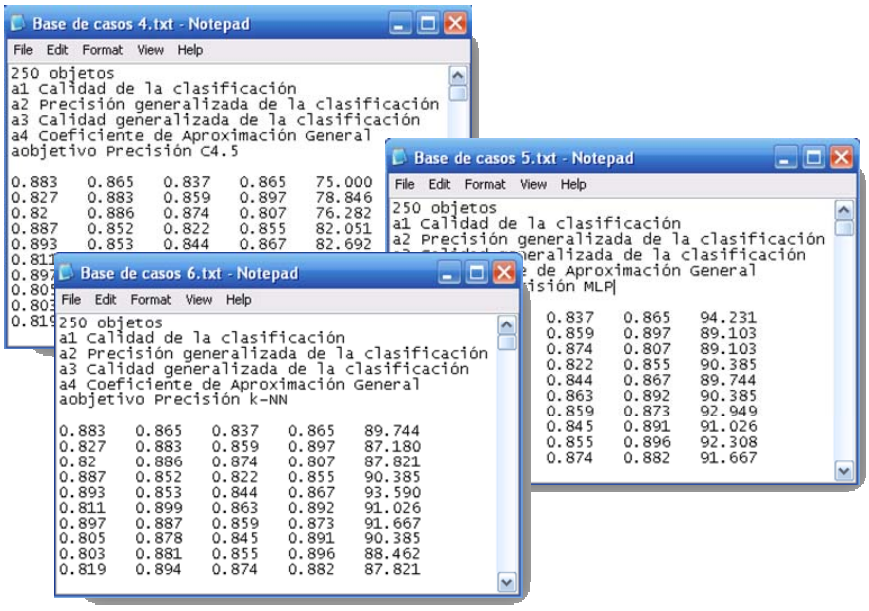

Figura A7.2 Conjuntos de datos construidos con los resultados de las medidas de inferencia y la precisión de los clasificadores