Universidad de Granada

Departamento de Ciencias de la Computación
e Inteligencia Artificial

Reducción de Datos basada en Selección Evolutiva de Instancias para Minería de Datos

Tesis Doctoral

José Ramón Cano de Amo

Granada, Julio de 2004

Universidad de Granada

Reducción de Datos basada en Selección Evolutiva de Instancias para Minería de Datos

MEMORIA QUE PRESENTA

José Ramón Cano de Amo

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Julio de 2004

DIRECTORES

Francisco Herrera Triguero y Manuel Lozano Márquez

Departamento de Ciencias de la Computación
e Inteligencia Artificial

La memoria titulada "*Reducción de Datos basada en Selección Evolutiva de Instancias para Minería de Datos*", que presenta D. José Ramón Cano de Amo para optar al grado de doctor, ha sido realizada dentro del programa de doctorado "*Diseño, Análisis y Aplicaciones de Sistemas Inteligentes*" del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los doctores D. Francisco Herrera Triguero y D. Manuel Lozano Márquez.

Granada, Julio de 2004

El Doctorando | Los Directores

Fdo: José Ramón Cano de Amo | Fdo: F. Herrera Triguero y M. Lozano Márquez

Tesis Doctoral parcialmente subvencionada por la Comisión
Interministerial de Ciencia y Tecnología con el proyecto
TIC2002-04036-C05-01

CICYT

TIC2002-04036-C05-01

Agradecimientos

Esta memoria esta dedicada a todas aquellas personas sin las cuales no hubiera sido posible.

Ante todo a mis padres, por que todo lo que se ha conseguido ha sido gracias a su cariño y apoyo y de los que estoy muy orgulloso. Esta memoria es por y para vosotros.

Si en el ámbito familiar he tenido suerte por el aliento y ánimo recibidos, ésta no ha sido menor con respecto a mis directores de tesis. Ambos, tanto Francisco Herrera como Manuel Lozano, han sido capaces de, mediante su paciencia, dedicación e inestimables consejos, ayudarme a llevar a buen puerto este viaje.

Por supuesto no me podría olvidar de todas aquellas personas que han estado a mi lado, y no ha sido fácil, durante todo este camino. Muchas gracias a Rafael Alcalá por toda la ayuda proporcionada, y así como Oscar Cordón y Jorge Casillas por servirme de modelo. Gracias Iñaki por las pequeñas charlas telefónicas.

Quiero así mismo expresar mi gratitud a aquellos compañeros que me han acompañado en mi peregrinar andaluz y me han suavizado los rigores de la distancia. De entre mis onubenses favoritos citar a Francisco Márquez, quién me hizo sentir como en casa, muchas gracias. Pero no fue el único, allí estuvieron Antonio Peregrín, Alfredo Sainz, Estefanía Cortés y el más recreativo, Manuel de la Villa. De Córdoba, citar a Sebastián Ventura por su cariñoso recibimiento y su comprensión. Y de Jaén, a mis compañeros Mari Lina y Jose María por hacer agradable el trabajo diario.

No quiero dejar de mencionar a los amigos por lo vivido y lo que nos queda por vivir: Fernando, Pedro, Moncho, Manolo, y al resto del equipo de Linares, que han reacogido al hijo pródigo.

Mi agradecimiento a todas aquellas personas que no por no citarlas han sido

menos importantes para el término de esta memoria. Quiero dedicaros el esfuerzo de este nuestro trabajo.

GRACIAS A TODOS

Índice

Índice de figuras

Índice de tablas

Introducción

A Planteamiento

En la actualidad, la extraordinaria evolución que se ha producido tanto a nivel de computación como de intercambio y recopilación de información ha ofrecido ingentes cantidades de datos para su estudio. Conforme crece su volumen, aumenta la dificultad para comprenderlos y tomar decisiones a partir de ellos.

A través del procesamiento de los datos se pretende obtener información útil y de calidad que podrá aplicarse a diferentes disciplinas de la ingeniería y las ciencias. La disciplina de investigación que estudia la extracción de conocimiento útil se denomina *Descubrimiento de Conocimiento en Bases de Datos* (en inglés Knowledge Discovery in Databases, con el acrónimo `KDD` que será empleado a lo largo de la memoria por ser estándar su uso) [HMS01]. Los modelos obtenidos en el marco de este área de trabajo se emplean para resolver problemas reales de información comercial, procesos industriales o de información científica, tales como análisis de ventas, detección de fraudes, planificación, gestión de redes, análisis de experimentos, etc. De esta forma, el desarrollo de modelos adecuados puede conducir a la obtención de decisiones exitosas en su ámbito de aplicación.

El proceso de descubrimiento de conocimiento está compuesto por una serie de etapas, siendo la *Minería de Datos* (`MDD`) la encargada de extraer modelos a partir de la información recogida [HRF04].

Dependiendo del objetivo perseguido, se pueden generar modelos siguiendo diferentes enfoques:

- Predictivos. Los modelos se generan con el objetivo de conseguir las mayores

capacidades de predicción sobre la base de datos.

- Descriptivos. Los modelos persiguen aportar conocimiento descriptivo sobre el problema en cuestión, descubriendo reglas de asociación, patrones interesantes entre los datos, etc.

  Un modelo descriptivo reciente consiste en el *Descubrimiento de Subgrupos*, donde se pretende generar modelos basados en reglas cuya finalidad es descriptiva, empleando una perspectiva predictiva para obtenerlos [LKFT04]. Su idea se basa en, dada una población de individuos (patrones de la misma clase) y una propiedad de esos individuos en la que estamos interesados, buscar subgrupos en esa población que sean estadísticamente "más interesantes", siendo tan grandes como sea posible y ofreciendo el mayor valor de atipicidad estadística con respecto a la propiedad en la que estamos interesados.

En esta memoria estamos interesados en la obtención de modelos desde dos perspectivas:

- Predictiva.
- Descriptiva basada en el descubrimiento de subgrupos.

Las técnicas de `MDD` son sensibles a la calidad de la información sobre la que se pretende extraer conocimiento. Cuanto mayor sea esta calidad, mayor será la de los modelos de toma de decisiones generados. En este sentido, la obtención de información útil para ser posteriormente procesada es un factor clave. Aparece por tanto en el proceso de descubrimiento una etapa de preprocesamiento de datos previa a la `MDD` [Pyl99].

Podemos considerar como *Preprocesamiento o Preparación de Datos* a todas aquellas técnicas de análisis de datos que permiten mejorar la calidad de los mismos, de modo que los métodos de `MDD` puedan obtener mayor y mejor información [ZZY03].

La relevancia de la preparación de los datos se debe a que:

- Los datos reales pueden ser impuros, pudiendo conducir a la extracción de modelos poco útiles. Dicha circunstancia puede estar originada por datos incompletos, datos con ruido o datos inconsistentes [KCH$^+$03].

- La preparación de los datos puede generar un conjunto de menor tamaño que el original, lo cual puede mejorar la eficiencia en MDD. En este aspecto, se pueden desarrollar tareas dirigidas a seleccionar datos relevantes, eliminar registros duplicados, anomalías, o bien reducir el volumen de datos, mediante la selección de características, de instancias, discretización, etc.

- La preparación origina datos de calidad, los cuales pueden conducir a modelos de calidad. Para ello, se emplean mecanismos que recuperan información incompleta, resuelven conflictos o bien, eliminan datos erróneos.

En esta memoria, de entre las diferentes estrategias a seguir en el preprocesado de datos, vamos a dirigir nuestra atención hacia la *Reducción de Datos* (RDD), donde el objetivo es extraer del conjunto original de datos un conjunto de datos más pequeño y representativo para confeccionar el modelo. Así mismo, la reducción se puede llevar a cabo de múltiples formas. En esta memoria nos centraremos en la *Selección de Instancias* (SII) donde se escogen las muestras más significativas del conjunto de datos [LM01a]. Más concretamente, en la *Selección de Prototipos* (SPP) empleando la regla del vecino más cercano.

El proceso de selección de instancias se puede orientar desde dos perspectivas posibles:

- Obtener un clasificador basado en la técnica del vecino más cercano vía la SPP. Se pretende aumentar la precisión del clasificador que utiliza la regla del vecino más cercano mediante la SII.

- La Selección de Conjuntos de Entrenamiento. Donde se considerará la calidad de los conjuntos obtenidos para la extracción de modelos mediante técnicas de MDD. Las medidas de calidad consideradas para valorar los subconjuntos dependerán del ámbito al que se dirijan los modelos generados. En nuestra memoria, los modelos pueden ser predictivos, en cuyo caso la medida es la precisión e interpretabilidad, o bien descriptivos orientados al descubrimiento de subgrupos, donde se valoraran factores tales como la atipicidad de la información encontrada, relevancia, etc.

Los *Algoritmos Evolutivos* (AAEE), son estrategias de optimización y búsqueda estocástica basadas en la selección natural y la genética [BFM97, Gol02]. La capacidad demostrada por los AAEE para explorar y explotar dominios de información extensa y de los que hay poco conocimiento, ha favorecido su aplicación en el

ámbito de la extracción de conocimiento. Se han planteando múltiples modelos de aprendizaje basados en AAEE considerándolos como problemas de optimización y búsqueda [Fre02]. En esta memoria estudiaremos su utilización como herramienta de selección de instancias para la RDD.

Los algoritmos de SII se ven afectados por el tamaño del conjunto de datos sobre el cual se aplican. Dado que las técnicas de SII presentan órdenes de eficiencia superiores a $O(n^2)$, siendo n el número de muestras del conjunto, los requerimientos tanto de tiempo de cálculo como de recursos necesarios aumentan considerablemente con el tamaño del conjunto de entrada. En el caso de los AAEE, dicho efecto es objeto de estudio en esta memoria, evaluándose el problema de escalado, esto es, analizar el comportamiento de los algoritmos cuando aumenta el tamaño de la base de datos.

En esta memoria, nos centraremos en el análisis del uso de los AAEE para la SII. Consideraremos el estudio desde la doble perspectiva de la SPP para clasificar con la técnica del vecino más cercano, y la selección de conjuntos de entrenamiento para extraer modelos. Esta segunda vía se puede emplear para obtener modelos con mayor capacidad de predicción e interpretabilidad, o bien para extraer modelos descriptivos en descubrimiento de subgrupos con índices elevados de atipicidad, relevancia, etc. Conjuntamente, analizaremos el problema de escalado de los algoritmos cuando aumenta el tamaño de la base de datos.

B Objetivos

Como se acaba de mencionar, el principal objetivo de esta memoria es analizar la selección evolutiva de instancias en la RDD desde una doble perspectiva. La SPP para clasificar según la regla del vecino más cercano y la generación de modelos predictivos y descriptivos de calidad, mediante la selección de conjuntos de entrenamiento, donde la calidad de los modelos podrá evaluarse en relación a la precisión e interpretabilidad o la atipicidad, novedad, etc., de las reglas que lo componen.

En concreto, el objetivo propuesto se subdivide en los siguientes objetivos concretos:

- *Evaluar la aplicación de los* AAEE *en la* SII. Analizamos el empleo de AAEE

que ponderen de forma adecuada la reducción del conjunto de datos con la menor pérdida de precisión. Se analizará el comportamiento de las diferentes técnicas de SII.

- *Analizar el problema de escalado en* SII. Los algoritmos de SII van a verse penalizados por el tamaño del conjunto de datos sobre el que se aplican. Se pretende estudiar en qué medida se produce esta penalización y ofrecer un mecanismo alternativo que la solvente. Para ello, se propone la combinación de la SII con la estratificación del conjunto de datos inicial, con la posterior combinación de los resultados por estrato.

- *Estudiar el comportamiento de los algoritmos de* SPP *para generar un clasificador basado en la regla del vecino más cercano.* En este caso, se analizarán los resultados ofrecidos por los algoritmos considerando la reducción y la capacidad de precisión del conjunto de prototipos obtenido a partir de su selección.

- *Analizar los algoritmos de* SPP *desde la perspectiva de la selección de conjuntos de entrenamiento para la obtención de modelos predictivos e interpretables.* Se estudia el modelo generado a partir del conjunto seleccionado en base a la capacidad de predicción y al tamaño que ofrece.

- *Evaluar los algoritmos de* SPP *desde la perspectiva de la selección de conjuntos de entrenamiento para la obtención de modelos descriptivos basados en el descubrimiento de subgrupos.* En este caso, se analiza el modelo en base a la interpretabilidad, novedad de la información que aporta, etc.

C Resumen

Para desarrollar los objetivos planteados, la memoria está organizada en cuatro capítulos y una sección de comentarios finales. La estructura de cada una de estas partes se introduce brevemente a continuación.

En el Capítulo 1, se lleva a cabo una breve revisión de conocimientos que permiten situar la presente memoria en su contexto. Estudiamos el KDD, analizando cada una de las etapas que lo componen. Profundizamos en la fase de preprocesamiento, presentando las diferentes vías que se pueden seguir en él. Nos

centramos en la RDD, estudiando sus estrategias de aplicación. De entre todas, dirigimos nuestra atención al estudio de la SII, concretamente, a la SPP basada en la regla del vecino más cercano. Finalmente, se introduce el empleo de los AAEE en el ámbito de la extracción de conocimiento, sirviéndonos de referencia para su empleo en los siguientes capítulos.

En el Capítulo 2, se presenta la selección evolutiva de instancias en la RDD desde dos perspectivas: SII para clasificación mediante el vecino más cercano y selección de conjuntos de entrenamiento para obtener modelos predictivos, utilizando árboles de decisión y en particular C4.5 [Qui93]. En un primer momento, se presentan los diferentes métodos de SPP no evolutivos que se van a estudiar. A continuación, se describen los AAEE empleados, destacando sus características principales. Se incluye el estudio experimental desarrollado, donde se especifica la metodología seguida, sus resultados y un completo análisis de los mismos.

En el Capítulo 3, se estudia el problema de escalado en la SII basada en clasificación empleando la regla del vecino más cercano. Para solventar el problema del incremento de tamaño de los conjuntos de datos se propone la combinación de estratificación del conjunto de datos con las técnicas de SII. El estudio nos permite comparar las técnicas evolutivas y no evolutivas siguiendo el modelo estratificado, analizando sus resultados y comportamiento.

El Capítulo 4 estudia la SII basada en selección de conjuntos de entrenamiento para grandes conjuntos de datos. La selección de conjuntos de entrenamiento se emplea para la generación de modelos predictivos y modelos descriptivos siguiendo la perspectiva del descubrimiento de subgrupos. Se estudian los modelos generados en ambos dominios previamente citados incluyendo el estudio experimental, la metodología seguida, los resultados obtenidos y el análisis de los mismos.

Incluimos una sección de "Comentarios Finales", que resume los resultados obtenidos en esta memoria, presentando algunas conclusiones sobre éstos. Finalmente, se comentarán algunos aspectos sobre trabajos futuros que quedan abiertos en la presente memoria.

Capítulo 1

Extracción de Conocimiento, Reducción de Datos y Selección de Instancias

En la actualidad la sociedad se enfrenta al reto de trabajar con volúmenes de información cada vez mayores. El KDD es un área de la computación que intenta explotar la ingente cantidad de información mediante el descubrimiento de patrones representativos y útiles, extrayendo conocimiento que pueda asistir a un humano para llevar a cabo tareas de forma más eficiente y satisfactoria. Fayyad et al. [FPSS96] define el proceso de KDD como *el proceso no trivial de identificación de patrones válidos, originales, potencialmente útiles y comprensibles en los datos.*

El KDD está compuesto por una serie de etapas, siendo la MDD la encargada de extraer modelos a partir de la información recogida [HRF04].

Debido al tamaño de las bases de datos, a la presencia de ruido, datos inconsistentes, redundantes, etc., se hace necesaria la aplicación de técnicas de preprocesamiento sobre los conjuntos de datos [Pyl99]. El objetivo perseguido por el preprocesamiento es obtener conjuntos de datos tales que al aplicar técnicas de MDD sobre ellos se generen modelos representativos con mayores prestaciones.

Nosotros centramos la atención en una de las técnicas de preprocesamiento de datos, la RDD y lo haremos via la SII.

El objetivo de este capítulo es introducir el proceso de KDD, deteniéndonos en particular en las etapas que lo componen. Dirigiremos la atención a la etapa de preprocesamiento, donde destacaremos la RDD. Dentro de éste ámbito, nuestro interés se dirigirá a la SII, y más concretamente a la SPP. En este capítulo estudiaremos las diferentes formas de llevarla a cabo y le dedicaremos especial atención al empleo de los AAEE.

Este capítulo de la memoria se organiza como sigue. En la Sección 1.1, se introduce la extracción de conocimiento. La Sección 1.2 describe el KDD, con las diferentes etapas que lo conforman. En la Sección 1.3, se estudia la MDD junto con las diferentes posibilidades de llevarla a cabo. Dentro del proceso de descubrimiento de información, nuestro interés se centra en el preprocesamiento, siendo este destacado en la Sección 1.4. La Sección 1.5 dirige su interés hacia la RDD como mecanismo de preprocesamiento, resumiendo las diferentes vías que se pueden seguir en su aplicación. En la Sección 1.6, se destaca la SII como estrategia de RDD, y se estudian los diferentes métodos que se pueden emplear, siendo la SPP uno de ellos y donde vamos a centrar el interés en la Sección 1.7. Finalmente, en la Sección 1.8, se introducen los AAEE aplicados a la extracción de conocimiento.

1.1. Introducción a la Extracción de Conocimiento

El aumento del volumen y variedad de información ha crecido de forma espectacular en los últimos años. La cantidad de información es tal que es necesario tratarla para poder utilizarla adecuadamente. Los datos tal cual se almacenan suelen ser procesados previamente para proporcionar beneficios directos. Su valor real reside en la información que podamos extraer de ellos: información que nos ayude a tomar decisiones o a mejorar nuestra comprensión de los fenómenos que nos rodean.

El método tradicional de convertir los datos en conocimiento consiste en un análisis e interpretación realizado de forma manual. Sin embargo, cuando la cantidad de datos de los que disponemos aumenta, el uso de herramientas de extracción de conocimiento y MDD es esencial [HRF04].

En la aplicación de técnicas de KDD (ver Figura 1.1), un gran esfuerzo se

dedica a la preparación de los datos [Pyl99, ZZY03]. Los métodos de MDD son muy potentes en la búsqueda de información de interés y pueden serlo aun más conforme se adecúan los datos mediante el preprocesamiento.

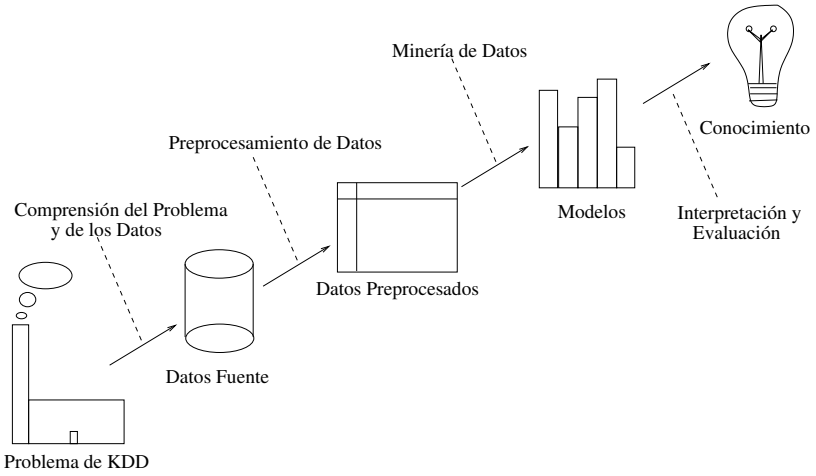

Figura 1.1: Etapas de procesamiento de la información

Como razones para la preparación de los datos podemos citar las siguientes [Pyl99]:

- Ajuste del tamaño del conjunto de datos para poder ser evaluado por técnicas de MDD.
- Ajuste del formato de los datos para poder ser evaluado por técnicas concretas de MDD.
- Gestión de datos perdidos.
- Tratamiento de ruido en la información.
- Eliminación de información redundante.
- Eliminación de datos inconsistentes.

El proceso de KDD se divide en una serie de etapas, como aparece reflejado en la Figura 1.1. Tanto el número de etapas como la función que se desempeña en cada una de ellas varía de una descripción de KDD a otra. En la siguiente sección analizaremos el proceso de KDD siguiendo el modelo descrito en [CCK+99].

1.2. El Proceso de Descubrimiento de Conocimiento en Bases de Datos o KDD

La metodología CRISP-DM (*Cross-Industry Standard Process for Data Mining* [CCK+99]) estructura el ciclo de vida de un proyecto de MDD en seis fases, que interactúan entre ellas de forma iterativa durante el desarrollo del proyecto, según podemos ver en la Figura 1.2:

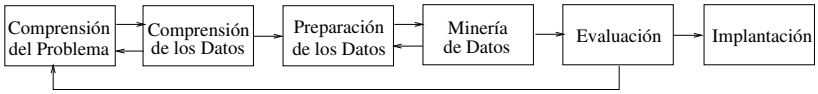

Figura 1.2: Fases de KDD según el modelo iterativo CRISP-DM

Normalmente los proyectos de MDD no terminan en la implantación del modelo, sino que se deben documentar y presentar los resultados de manera comprensible para lograr así un aumento del conocimiento. Además en la fase de explotación se debe de asegurar el mantenimiento de la aplicación y la posible difusión de los resultados [FPSS96].

A continuación describiremos en qué consiste cada una de esas etapas.

- Comprensión del problema: Esta primera etapa se centra en la comprensión del problema y en concretar los objetivos perseguidos, para así convertir este conocimiento en la definición de un problema de KDD. Para delimitar los objetivos, al finalizar esta fase sería necesario:

  - Obtener un conocimiento adecuado del problema.
  - Tener descritos claramente los objetivos.
  - Establecer el criterio de éxito o utilidad que se desea alcanzar.

  En esta fase se diseñará el plan para alcanzar dichos objetivos. Esta información sería el punto de partida para la siguiente etapa dentro del KDD.

- Comprensión de los datos: En esta fase se comienza a trabajar con el conjunto inicial de datos. Se llevan a cabo diversas actividades tales como:

  - Familiarizarse con la información.
  - Identificar problemas en la calidad de los datos.

- Detectar subconjuntos interesantes para formular hipótesis sobre información oculta.

Tras estudiar el conjunto inicial de datos, se efectúa una descripción de dichos datos, desarrollando una exploración general sobre los mismos para buscar información oculta. Finalmente, se verifica la calidad de los datos.

- Preparación de los datos: Tomando como partida la información recogida en la etapa anterior en ésta fase se desarrollan actividades destinadas a confeccionar el conjunto de datos final (conjunto que servirá de entrada al algoritmo de `MDD`) a partir del conjunto inicial. Las tareas dedicadas a la preparación de los datos se pueden aplicar repetidas veces, sin tener por qué utilizarse en un orden concreto. Al final de esta etapa tendremos el conjunto de datos preparado para ser utilizado en `MDD`.

  Nuestro interés en la tesis se centrará en las técnicas de `RDD` como mecanismos de preparación o preprocesado.

- Minería de datos: En esta etapa se efectúa la extracción de modelos que aportan conocimiento sobre los datos procesados. Normalmente se pueden emplear diferentes tipos de técnicas sobre el mismo problema.

- Evaluación: Llegados a esta fase, ya tenemos una técnica desarrollada de la que se ha evaluado la calidad de sus soluciones. Antes de proceder en la última etapa a la implantación de la solución para su uso habitual, es importante revisar concienzudamente tanto la técnica como las resultados proporcionados. Habría que revisar los pasos llevados a cabo en la aplicación del algoritmo para asegurarnos que se adecúan, tanto el método como las soluciones, a los objetivos perseguidos. Al final de esta fase se habrá decidido la utilización de los resultados obtenidos con la técnica de `MDD`.

  Al evaluar los resultados habrá que considerar tanto la exactitud como la capacidad de generalización de la técnica.

- Implantación: La utilización del algoritmo no es generalmente el punto final del proceso de `KDD`. Incluso si el propósito del desarrollo es obtener conocimiento sobre los datos, el conocimiento ganado deberá ser organizado y preparado para poder ser utilizado. Dependiendo de los requerimientos, el proceso de implantación puede ser tan simple como la generación de un informe o tan complejo como construir un producto software comercial.

1.3. Minería de Datos

En la MDD se integran un conjunto de áreas que tienen como propósito la identificación del conocimiento obtenido a partir de las bases de datos que aportan un sesgo hacia la toma de decisiones [WF00, HMS01, HRF04].

El primer paso dentro del proceso de MDD es decidir que técnica se va aplicar para efectuar la búsqueda de información. El siguiente paso consiste en estudiar la calidad y validez de esa técnica en cuanto a la resolución que ofrece del problema. Tras asegurarnos que es la adecuada, procedemos a aplicarla y evaluar sus parámetros para ajustarlos. A continuación, obtendremos diversas soluciones del problema que serán analizadas antes de avanzar a la siguiente etapa. La evaluación se lleva a cabo en este momento para realimentar el conocimiento sobre la técnica aplicada, con la idea de refinar ajustes o corregir errores, todo ello antes de avanzar a la siguiente fase dentro del proceso de KDD.

Nuestro estudio se centra en llevar a cabo el procesado mediante la reducción del conjunto inicial. Se pretende con ello mejorar el comportamiento del algoritmo de MDD. Por tanto, dada la importancia de esta etapa vamos a describir a continuación en qué consiste y las diferentes técnicas que se aplican para llevarla a cabo.

El nombre de MDD se deriva de las similitudes entre buscar valiosa información en grandes bases de datos y minar una montaña para encontrar una veta de metales valiosos. Ambos procesos requieren examinar una inmensa cantidad de material, o investigar inteligentemente hasta encontrar exactamente donde residen los valores de interés.

Algunas de las técnicas más comúnmente usadas son:

- Redes neuronales artificiales: Modelos predecibles no-lineales que aprenden a través del entrenamiento y semejan la estructura de una red neuronal biológica.
- Modelos estadísticos: La técnica anterior se caracteriza por emplear un único atributo como base para la toma de decisiones y elige aquel que mejor funciona. Otra forma de abordar el problema consiste en utilizar todos los atributos y permitirles contribuir en la decisión, ya que todos son igualmente importantes y independientes unos de otros. Este método está basado en la regla de Bayes de probabilidad condicionada.

- Modelos lineales: Los modelos basados en reglas y árboles de decisión funcionan adecuadamente cuando se trabaja con atributos nominales. Pueden ser extendidos para emplear valores numéricos, sin embargo, hay esquemas que funcionan de forma más natural con atributos que son numéricos. La regresión linear como uno de estos modelos es la técnica más adecuada a considerar cuando el valor de salida o la clase buscada es un valor numérico y el resto de atributos son numéricos también.
- Razonamiento basado en casos: Para predecir una situación futura, o para tomar la decisión correcta, este tipo de sistemas buscan la situación pasada que se asemeja más a la que está aconteciendo, empleando la predicción o decisión que se tomó entonces. Las predicciones están basadas en datos históricos. Se necesita una función para medir la semejanza entre diferentes casos para poder decidir a cual se asemeja más.
- Inferencia empleando reglas. Dependiendo del objetivo perseguido las diferentes técnicas se pueden clasificar:
    - Inducción Predictiva: Ámbito en el que podemos destacar el aprendizaje supervisado mediante reglas de clasificación. Para la construcción de las reglas de clasificación los algoritmos que generan los árboles de decisión siguen una estrategia de divide y vencerás, generando el árbol de arriba hacia abajo. Esto también se puede llevar a cabo en sentido contrario. Se puede tomar una clase y buscar la regla que cubre a todas las instancias de esa clase, excluyendo al mismo tiempo a aquellas instancias que no pertenecen a ella.
    - Inducción Descriptiva: Donde se destaca el descubrimiento de reglas de asociación siguiendo un modelo no supervisado de aprendizaje. El objetivo perseguido es encontrar reglas individuales que definan patrones interesantes en los datos. Las reglas de asociación se pueden obtener ejecutando un procedimiento de inducción de reglas de tipo divide y vencerás. Sin embargo, un atributo de la parte derecha de la regla puede presentar cualquier valor y una única regla de asociación puede predecir en ocasiones el valor de más de un atributo. Para encontrar estas reglas habría que ejecutar el procedimiento de inducción de reglas una vez por cada posible combinación de atributos, con cada posible combinación de valores en la parte derecha de la regla. Un nuevo mecanismo de inducción descriptiva es el denominado Descubrimiento de Subgrupos. En este dominio se lleva a cabo la búsqueda

de subgrupos en el conjunto de datos que sean estadísticamente "más interesantes", siendo tan grandes como sea posible y ofreciendo el mayor valor de atipicidad estadística con respecto a la propiedad en que estemos interesados. Se basa en obtener modelos descriptivos mediante el empleo de mecanismos predictivos [LKFT04].

1.4. Preparación de los Datos

Como ya se ha comentado, la preparación de los datos es la tarea que más tiempo consume dentro del KDD. Se pretende obtener un conjunto de datos de calidad tal, que al emplearlo como entrada en MDD puede conducir a obtener patrones o reglas de mayor calidad [Pyl99].

La importancia de la preparación de los datos se ve reflejada en los tres aspectos siguientes [ZZY03]:

- Los datos del mundo real puede ser incompletos, inconsistentes, o presentar ruido.
- La preparación genera conjuntos de datos que son menores que el conjunto original, lo que puede mejorar significativamente la eficiencia del algoritmo de MDD.
- La preparación da lugar a datos de calidad, al recuperar instancias incompletas, corregir errores o resolver conflictos.

Para mejorar la calidad de los conjuntos de datos, desarrollando cualquiera de las tareas anteriormente citadas, se pueden seguir las estrategias que describimos a continuación y que aparecen en la Figura 1.3:

- Limpieza de datos: Se aumenta la calidad de los datos al nivel requerido mediante técnicas de análisis selectivo. Este proceso consiste en la eliminación de datos erróneos o inconsistentes [KCH+03, COMV04].
- Reducción de datos: Consiste en decidir que datos deben ser utilizados para el análisis. El criterio que se sigue incluye la relevancia con respecto a los objetivos que se persiguen en la MDD, y limitaciones técnicas tales como pueden ser volúmenes máximos de datos o bien tipos de datos concretos.

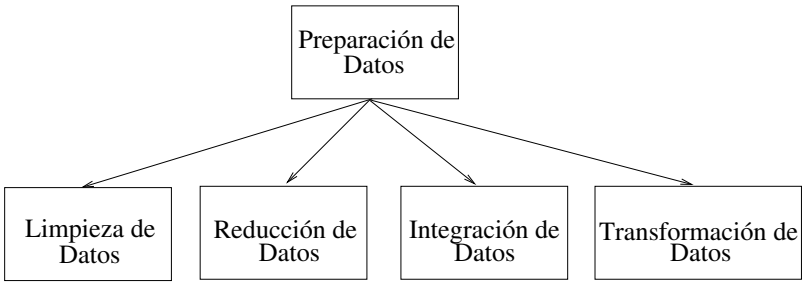

Figura 1.3: Estrategias para el preprocesado de datos

Nuestro estudio lo enfocaremos desde esta perspectiva del preprocesamiento: reducir el volumen de datos seleccionando los más relevantes para su posterior uso por algoritmos de MDD [LM01b].

- Integración de datos: Se basa en combinar múltiples tablas o registros para crear nuevos registros o valores. El combinar tablas se refiere a unir dos o más tablas que presentan diferente información sobre los mismos objetos. La combinación de datos también incluye la agregación. La agregación consiste en operaciones donde se obtienen nuevos valores mediante la unión de información de varios registros o tablas. Esta tarea comprende así mismo operaciones relativas a construcción de datos tales como la producción de atributos derivados, nuevas muestras completas, o transformaciones de los valores de atributos ya existentes. Los atributos derivados se pueden construir con uno o más atributos presentes en el mismo patrón [DDBM03, SSS04].

- Transformación de datos: Las transformaciones consisten principalmente en modificaciones sintácticas llevadas a cabo sobre los datos, sin que supongan un cambio en el significado de los mismos. Estas transformaciones pueden ser necesarias para la técnica de MDD aplicada [Lin02].

Las estrategias anteriormente descritas no son mútuamente excluyentes. Existen técnicas de preprocesado que podrían seguir dos o más de las vías indicadas y habría que clasificarlas como una combinación de ambas (por ejemplo, la compactación de datos, que reduce e integra).

En nuestro estudio, centraremos la atención en las técnicas de preprocesado basadas en la RDD.

1.5. Reducción de Datos

Las técnicas de MDD, sobre todo aquellas basadas en instancias, tienden a almacenar muchas, en algunos casos todas, las instancias que se le presentan durante el entrenamiento, lo que provoca inconvenientes adicionales tales como:

- Se aumenta el tiempo de respuesta. Cuantos más ejemplos se almacenen, mayor será el tiempo necesario para clasificar casos no vistos. Ante un ejemplo que se desee clasificar, estos algoritmos buscan aquel de entre los que tienen almacenados que más se parece al caso presentado. La clasificación que se le dará a este último será la clase de aquel.
- Se aumenta la sensibilidad al ruido y la posibilidad de sobreajuste en el conjunto de entrenamiento. Al guardar un mayor número de instancias, es más probable que se retengan ejemplos ruidosos. Eso provocará que esas instancias de escasa calidad clasifiquen incorrectamente todos los casos que caigan dentro de su región de decisión.
- Al retener demasiados ejemplos de entrenamiento, la solución obtenida es poco comprensible para la mente humana. Difícilmente un ser humano puede comprender la solución de un problema que emplea cientos de ejemplos para representarla. Cuantos menos ejemplos formen la solución dada, más comprensible será.

Se hace por tanto necesario un preprocesamiento previo en el que se disminuya el tamaño del conjunto almacenado, objetivo de la RDD.

La RDD se puede llevar a cabo empleando las diferentes técnicas que aparecen en la Figura 1.4:

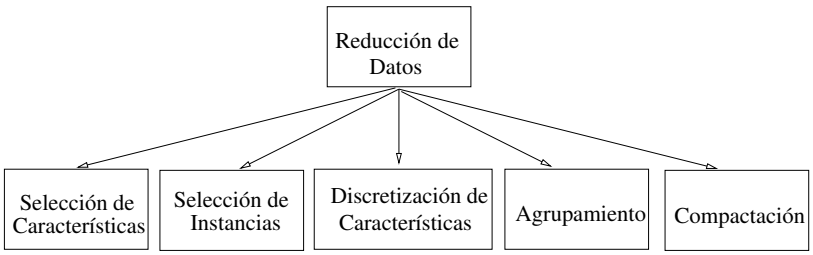

Figura 1.4: Técnicas de reducción de datos

Pasaremos a describir cada una de ellas en las siguientes secciones.

1.5.1. Selección de Características

La mayoría de algoritmos de aprendizaje automático están diseñados para aprender cuales son los atributos más apropiados para tomar decisiones. Por ejemplo, los árboles de decisión eligen el atributo más prometedor para llevar a cabo la división en cada nodo interno, y nunca deberían seleccionar - en teoría - atributos irrelevantes o carentes de utilidad.

En principio podríamos suponer que un aumento en el número de atributos incrementaría también la capacidad de discrimización, pero lo que sucede es el hecho contrario. Si en algún punto en el que se está generando el árbol de decisión se escoge un atributo irrelevante, se introducen errores aleatorios cuando el conjunto de test es procesado. Esta situación es debida a que conforme se va profundizando en el árbol, menor es la cantidad de datos disponibles para decidir la selección. En un punto con pocos datos, un atributo irrelevante podría ser seleccionado como candidato para llevar a cabo la división. Debido a que el número de nodos crece exponencialmente con la profundidad, la posibilidad de escoger un atributo de este tipo se ve considerablemente aumentada.

Generadores de árboles de decisión del tipo Divide-y-vencerás, o bien generadores de reglas del tipo separa-y-vencerás adolecen de este problema debido a que inexorablemente reducen la cantidad de datos con los que toman sus decisiones. Los algoritmos de aprendizaje basados en instancias son muy susceptibles a atributos irrelevantes debido a que siempre trabajan tomando tan solo un conjunto

de instancias de entrenamiento para tomar sus decisiones. *Naive Bayes* ignora robustamente atributos irrelevantes [DP96]. Asume que todos los atributos son independientes unos de otros. Esto provoca el que *Naive Bayes* emplee atributos redundantes.

Debido al efecto negativo de atributos irrelevantes en la mayoría de esquemas de aprendizaje automático, es común llevar a cabo un proceso de selección de atributos previo al aprendizaje [LM98a, LM98b]. El proceso clásico de selección de características aparece reflejado en la Figura 1.5.

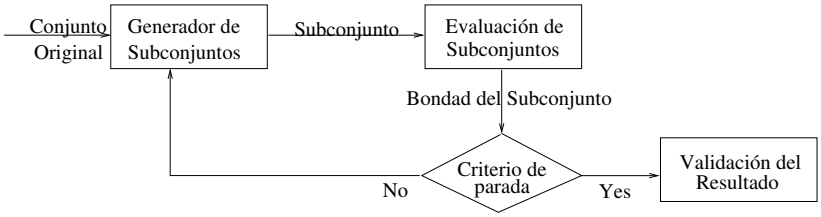

Figura 1.5: Proceso de selección de características

A continuación vamos a presentar diferentes técnicas empleadas para efectuar la selección de características. Una posible forma de clasificar estas técnicas es basarnos en el mecanismo de selección empleado. Tenemos dos aproximaciones: filtro y cobertura. Los métodos basados en filtro desarrollan la selección considerando características generales de los datos. Las estrategias basadas en envoltura emplean algoritmos de `MDD` para decidir su selección, siendo ese método el que se empleará posteriormente para `MDD` con el subconjunto seleccionado.

En este estudio se clasificarán los algoritmos basándonos en las principales características que son propias de los algoritmos de selección de esta naturaleza: medida de evaluación, estrategia de búsqueda y dirección de búsqueda.

La medida de evaluación es la medida empleada para valorar la bondad del conjunto seleccionado. Se pueden emplear tres tipos diferentes:

- Clásica, con medidas tales como por ejemplo la distancia, ganancia de información o bien medidas de dependencia entre características;
- Acierto, siendo la medida del acierto conseguido al clasificar empleando un determinado subconjunto de instancias;
- Consistencia, considerando como medida de este tipo a la medida de incon-

sistencia, de tal forma que inconsistencia cero significa consistencia total. El nivel de inconsistencia se calcula considerando que dos instancias son inconsistentes si se diferencian tan solo en su etiqueta de clase.

La estrategia de búsqueda representa las combinaciones de subconjuntos de características que serán evaluados hasta encontrar la solución final y puede ser de tres tipos:

- Completa, donde se cubren todas las combinaciones posibles de selección;
- Heurística, al reducir el número de combinaciones a evaluar basándose en la información disponible, aunque sea mínima;
- No determinista, con un tipo de búsqueda donde no se puede esperar la misma solución en cada ejecución. Se pretende con ello no perderse en mínimos locales y encontrar algunas interdependencias entre características que la búsqueda heurística es incapaz de capturar.

La dirección de búsqueda hace referencia al modo en el cual se va creando el conjunto de características seleccionadas. Se puede llevar acabo de tres formas:

- Búsqueda secuencial hacia adelante, donde se comienza con un conjunto vacío de características al que se le van añadiendo secuencialmente nuevas, una a una, procedentes del conjunto inicial hasta que se alcanza una condición de parada;
- Búsqueda secuencial hacia atrás, en la que se parte de un conjunto con todas las características del que se va eliminando secuencialmente una a una hasta que se satisface una condición de parada;
- Búsqueda aleatoria, esquema de búsqueda que produce conjuntos de características siguiendo un patrón aleatorio. De esta forma se evita la posibilidad de acabar en un óptimo local como le puede suceder a los dos esquema previos.

A continuación clasificaremos los diferentes algoritmos de selección de características según los tres componentes anteriores:

- Métodos de completitud: En este grupo encontramos aquellas técnicas que emplean búsqueda completa, cubriendo totálmente el espacio de búsqueda. Dentro de este conjunto de mecanismos de selección encontramos los siguientes (ver Tabla 1.1):

Tabla 1.1: Métodos de Completitud

Algoritmo	Estrategia	Dirección	Medida	Referencia
Focus	Completa	Adelante	Consistencia	[AD91]
Aut. Branch & Bound	Completa	Atrás	Consistencia	[LMD98]
Best First	Completa	Adelante	Clásica	[XYC88]
Beam Search	Completa	Adelante	Acierto	[Doa92]
Branch & Bound	Completa	Atrás	Clásica	[NF77]

- Métodos Heurísticos: Son técnicas caracterizadas por sacrificar la promesa del subconjunto solución óptimo por obtener una solución rápida. Para ello emplean el conocimiento disponible para dirigir la búsqueda. A continuación presentamos algunos de estos métodos (ver Tabla 1.2):

Tabla 1.2: Métodos de Heurísticos

Algoritmo	Estrategia	Dirección	Medida	Referencia
Wrap1	Heurístico	Atrás	Acierto	[LM98b]
SetCover	Heurístico	Adelante	Consistencia	[Das97]
SOAP	Heurístico	Adelante	Dependencia	[RRA02]

- Métodos Estocásticos: Este tipo de técnicas permiten la búsqueda del subconjunto de características óptimo mediante la generación aleatoria de subconjuntos. A continuación presentamos algunos de estos métodos (ver Tabla 1.3):

Tabla 1.3: Métodos Estocásticos

Algoritmo	Estrategia	Dirección	Medida	Referencia
Algor. Genéticos	No Determ.	Aleatorio	Cualquiera	[IML+01]
EDA	No Determ.	Aleatorio	Cualquiera	[ILS01]
Enfriamiento Simulado	No Determ.	Aleatorio	Cualquiera	[SS88]
Las Vegas Filter	No Determ.	Aleatorio	Consistencia	[LS96b]
Las Vegas Wrapper	No Determ.	Aleatorio	Acierto	[LS96a]

- Métodos Ponderando Características: Este tipo de técnicas se distinguen por no llevar a cabo ningún tipo de selección de forma explícita. En lugar de eso asocian a cada característica un valor de ponderación con el cuál podrán modificar su participación en el posterior proceso de aprendizaje automático. De entre estos métodos podemos destacar (ver Tabla 1.4):

Tabla 1.4: Métodos Ponderando Características

Algoritmo	Estrategia	Dirección	Medida	Referencia
Relief	Heurístico	Aleatorio	Clásico	[KR92]

- Métodos Híbridos: Con la hibridación de técnicas se pretende explotar las ventajas de unos métodos, eliminando sus inconvenientes. De entre estas técnicas podemos destacar (ver Tabla 1.5):

Tabla 1.5: Métodos Híbridos

Algoritmo	Estrategia	Dirección	Medida	Referencia
Quick Branch & Bound	No Determ.	Aleatorio	Consistencia	[DL98]

- Aproximación Incremental: Estas técnicas se basan en la idea de llevar a cabo la selección del subconjunto de características sin utilizar el conjunto completo de instancias de que se dispone. Se pretende con ello hacer frente al problema que aparece en los algoritmos de selección de características cuando se enfrentan a conjuntos de datos de elevado tamaño. Como técnica representativa habría que citar (ver Tabla 1.6):

Tabla 1.6: Aproximación Incremental

Algoritmo	Estrategia	Dirección	Medida	Referencia
Las Vegas Incremental	No Determ.	Aleatorio	Consistencia	[LS98]

1.5.2. Selección de Instancias

La reducción del conjunto inicial de datos mediante `SII` está basada en escoger tan solo las muestras más representativas de entre todo el conjunto, de tal forma que el conjunto seleccionado conserve las mismas prestaciones.

La Sección 1.6 está dedicada a describir este mecanismo de reducción.

1.5.3. Discretización de Características

Los atributos aparecen en las instancias en muchos casos en diferentes formatos: nominales, discretos y continuos. Los valores discretos representan intervalos dentro de un espectro continuo de valores. De esta forma, el número de valores continuos para un atributo puede ser infinito, mientras que el número de valores para un atributo discreto es frecuentemente de unos pocos o finito. La necesidad de discretizar los atributos puede venir impuesta por el algoritmo de aprendizaje que se emplee sobre ellos, que o bien no puede aplicarse sobre atributos continuos o bien es altamente ineficiente [LHTD02, ARBD04].

Como valor añadido habría que destacar que tanto para usuarios como expertos, los valores discretizados son más fáciles de entender y explicar. Así mismo, como refleja [DKS95], la discretización permite que el aprendizaje se lleve a cabo de forma más rápida, aumentando su precisión al mismo tiempo.

La discretización podríamos definirla como el proceso de cuantificar atributos continuos y podemos clasificar las diferentes técnicas de la siguiente forma:

- Combinación: Se sigue una estrategia de abajo hacia arriba. Consiste en comenzar con la lista completa de valores continuos utilizándolos como puntos de corte e ir eliminándolos mediante combinaciones sucesivas entre ellos conforme el proceso de discretización progresa. Para decidir que puntos combinar se emplea como medida χ^2. La medida χ^2 determina la semejanza de intervalos adyacentes basándose en su nivel de significancia. La idea es comprobar la hipótesis de que dos intervalos adyacentes deben ser independientes de la clase. Si son independientes, deben ser combinados, en otro caso permanecen separados. Según el mecanismo de discretización empleado, dentro de este grupo de técnicas tenemos (ver Tabla 1.7):

Tabla 1.7: Métodos de Discretización por Combinación

Algoritmo	Referencia
ChiMerge	[Ker92]
Chi2	[LS95]
ConMerge	[WL98]
USD	[GARR$^+$02]

- División: La estrategia seguida en este conjunto de técnicas es de arriba hacia abajo. De esta forma, se comienza con un conjunto vacío de puntos de corte y se van añadiendo nuevos por división del espacio continuo mientras el proceso progresa. Los diferentes métodos podemos clasificarlos según su naturaleza:

  - No Supervisada: Las técnicas no emplean la clase a la que pertenecen las instancias para discretizarlas. Las técnicas que aquí se agrupan comparten su medida de discretización, siendo por acumulación. Consiste en discretizar los atributos contínuos asignándoles un determinado numero de acumuladores o contenedores. El modo en el cual se creen dichos acumuladores dará lugar a las dos siguientes técnicas (ver Tabla 1.8):

Tabla 1.8: Métodos de Discretización por División No Supervisados

Algoritmo	Referencia
Igual Anchura	[CGB94]
Igual Frecuencia	[CGB94]

- Supervisada: Las técnicas aquí presentes se caracterizan por utilizar información sobre la clase a la que pertenece una instancia durante el proceso de discretización. Se pueden clasificar según la medida de discretización empleada en (ver Tabla 1.9):

Tabla 1.9: Métodos de Discretización por División Supervisados

Algoritmo	Medida	Referencia
ID3 - C4.5	Entropía	[Qui86, Qui93]
D2	Entropía	[Cat91]
Entropía-MDLP	Entropía	[FI93b]
Contraste	Entropía	[dM93]
Mantaras	Entropía	[CM97]
Khiops	Entropía	[Bou04]
Algoritmo de Fayyad e Irani	Entropía	[FI93a]
1R	Acumulación	[Hol93]
Entropía Marginal Máxima	Acumulación	[DKS95]
Zeta	Dependencia	[HS97]
Cuantización Adaptativa	Precisión	[CBS91]

1.5.4. Agrupamiento de Datos

Las técnicas de agrupamiento se aplican en aprendizaje no supervisado. En ellas no disponemos de clases a predecir y queremos separar las instancias en grupos. Los grupos obtenidos reflejan relaciones existentes entre las instancias que pertenecen a ellos.

El proceso típico de agrupamiento consta de las siguiente etapas [JD88]:

- Definición de la representación de las instancias. Habría que concretar el número de grupos a crear, el número de prototipos disponibles y el número, tipo y escala de las características de cada patrón.
- Definición de la medida de proximidad entre instancias según el dominio de los datos. Se pueden emplear diferentes tipos de medidas, como puede ser la distancia euclídea, que refleja diferencias entre prototipos. Otras medidas alternativas dependiendo del algoritmo empleado pueden ser la distancia de Mahalanobis, la distancia de Hausdorff, etc.
- Separación en los diferentes grupos. La obtención de los grupos se puede llevar a cabo de diferentes formas. Tras describir los pasos que componen la tarea del agrupamiento de datos, se ofrecerá una taxonomía de los métodos que se pueden aplicar para ello.
- Abstracción de los grupos obtenidos. Durante este proceso se extrae una representación simple y compacta del conjunto de datos. La simplicidad puede ser considerada desde la perspectiva del análisis automático o bien desde el punto de vista de la interpretabilidad humana. Comúnmente, esta abstracción consiste en una descripción compacta de cada cluster mediante un prototipo o bien mediante el empleo de centroides.
- Evaluación del resultado. Durante esta etapa se validan los agrupamientos obtenidos comprobando si se ajustan al comportamiento esperado.

Existen diferentes vías a seguir para llevar a cabo el agrupamiento de los datos. Según la estrategia seguida podemos clasificar a los métodos empleando la siguiente taxonomía [JMF99]:

- Agrupamiento Jerárquico: Existen una primera división en grupos a nivel mayor y se van refinando posteriormente cada uno de los grupos sucesivamente. Podemos destacar aquí a los siguientes métodos (ver Tabla 1.10):

Tabla 1.10: Métodos de Agrupamiento Jerárquico

Algoritmo	Referencia
Conexión Simple	[SS73]
Conexión Completa	[Kin67]
Mínima Varianza	[Mur84]
Algoritmos Genéticos	[LL99]

- Agrupamiento Particional: El algoritmo en este caso obtiene una partición única de los datos, en vez de una estructura de agrupamiento. Estas técnicas producen grupos mediante la optimización de funciones definidas local o globálmente. Dependiendo del criterio de optimización seguido las podemos clasificar según la Tabla 1.11:

Tabla 1.11: Métodos de Agrupamiento Particional

Algoritmo	Func. a Optimizar	Referencia
K-Medias	Error Cuadrático	[McQ67]
Isodata	Error Cuadrático	[BH67]
Agrupamiento Dinámico	Error Cuadrático	[Sym77]
Árbol minimal extensivo	Modelo Gráfico	[Zha71]

- Agrupamiento empleando la regla del vecino más cercano: Dado que la proximidad juega un papel clave en la noción intuitiva de grupo, la distancia según el vecino más cercano puede ser utilizada como base en los mecanismos de agrupamiento. Lu y Fu propusieron un procedimiento iterativo en [LF78].

- Agrupamiento difuso: En este caso cada instancia se asocia con cada uno de los grupos empleando una función de pertenencia [Zad65]. La primera aplicación de agrupamiento difuso la encontramos en [Rus69]. El libro de Bezdek es una buena fuente de material sobre el tema de agrupamiento difuso, donde podemos encontrar el algoritmo c-medias difuso [Bez81].

1.5.5. Compactación de Datos

Formálmente podemos describir la compactación de datos como un mecanismo de compresión que pretende conservar información estadística [DuM01]. Supongamos que el conjunto inicial de datos es una matriz `Y` compuesta por `n` filas o instancias y `m` columnas o características. El conjunto compactado sería la matriz `X` compuesta por `p` filas y `m`+1 columnas, donde `p` << `n`. La columna extra en `X` se trata de una columna de pesos w_i, i=1, ..., p, donde $w_i > 0$ y $\sum_i w_i = n$. La distribución n-dimensional de las filas de `X` ponderadas por w_i pretende aproximar la distribución de las filas de `Y` lo suficientemente bien que el análisis estadístico de `X` sea un sustituto aceptable del análisis deseado de `Y`.

Existen dos mecanismos triviales para efectuar la compactación de los datos y que suelen ser empleados como referencia en las comparativas [DuM01]:

- El primero de ellos consiste en un muestreo aleatorio, donde `X` consiste en un conjunto `p` aleatorio de muestras de `Y`, donde cada una tiene un peso asociado de $w_i = n/p$. El mayor inconveniente de este estrategia radica en imprecisión introducida por la varianza del muestreo.

- El segundo mecanismo de compactación podría ser denominado de extracción de filas singulares. En este caso, `X` consiste en un conjunto de filas insustituibles de `Y`, y w_i es la multiplicidad de la i-ésima fila de `X` en `Y`.

Podemos destacar los siguientes mecanismos de compactación de datos (ver Tabla 1.12):

Tabla 1.12: Métodos de Compactación de Datos

Algoritmo	Referencia
Comparación de Momentos sin Depósito	[DVJ+99]
Modelo basado en Probabilidad	[MRD+02]
Comparación empírica de Momentos de Probabilidad	[Owe03]

1.6. Selección de Instancias

La SII se enfrenta al problema de escoger las muestras más representativas de un conjunto determinado [KA87, BM02, LM02].

Disminuyendo el conjunto inicial de datos se consigue reducir tanto la complejidad en tiempo de cálculo, como los recursos de almacenamiento. La eliminación de instancias no tiene porqué producir una degradación de los resultados. Esto es debido a la existencia en la información de muestras repetidas o ruido. Es interesante el hecho de que cada instancia presente un cierto grado de libertad suficiente, tal que si reducimos su número podemos en algunos casos superar situaciones de sobreaprendizaje. La SII se puede llevar a cabo siguiendo diferentes vías, como podemos ver en la Figura 1.6.

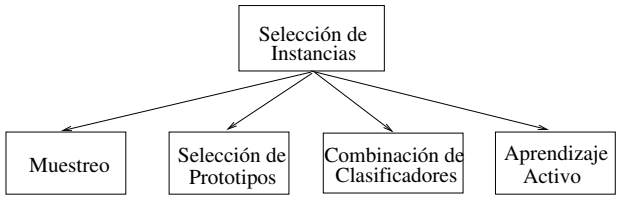

Figura 1.6: Estrategias de selección de instancias

Pasaremos a describir cada una de ellas a continuación:

- Muestreo: En Muestreo se escoge un subconjunto de instancias del conjunto original, mediante un proceso aleatorio de selección caracterizado por que cada muestra presenta una probabilidad de ser escogida.

  Diferentes modelos de muestreo existentes son:

  - Muestreo Aleatorio [Ska94, BFH01]: En este modelo de Muestreo, cualquier elemento del conjunto tiene la misma probabilidad de ser seleccionado. Como variantes aparecen el Muestreo Aleatorio con y sin Reemplazo. La diferencia entre ambos se refleja en que en el primero (con Reemplazo) una misma instancia puede ser seleccionada múltiples veces.
  - Muestreo Estratificado [BFH01]: Cuando la población está formada por un conjunto homogéneo de grupos, es conveniente y más efectivo el seleccionar elementos de cada uno de esos grupos. Para ello se divide

el conjunto en estratos no superpuestos, seleccionando a continuación una serie de muestras de cada uno de esos estratos. El conjunto final seleccionado estará formado por la unión de los muestreos en cada uno de los estratos.

- Muestreo por Agrupamiento [BFH01]: En caso de que la población esté compuesta por una serie de grupos, siendo cada uno de los cuales una "miniatura"del conjunto completo, es posible estimar correctamente las características de la población seleccionando el grupo más pequeño y todos sus elementos. Para aplicar esta idea, el conjunto inicial se divide en subpoblaciones mutuamente excluyentes denominadas "agrupamientos". A continuación se escogen algunos de estos agrupamientos, añadiendo todas las instancias que los forman al conjunto seleccionado final. A diferencia del Muestreo Estratificado, en este caso es deseable que las instancias en cada agrupamiento sean lo más heterogéneas como sea posible y todos los agrupamientos similares los unos a los otros presentando niveles de varianzas mínimos.
- Muestreo Sistemático [KR88, BFH01]: Se pretende con este Muestreo que todas las unidades del conjunto presenten las mismas oportunidades de ser escogidas. Supongamos que tenemos un conjunto inicial de tamaño `n` del que queremos seleccionar un Muestreo de tamaño `s`. Para llevarlo a cabo se seleccionará un número aleatorio entre 1 y `k` para a continuación seleccionar la instancia `k` del conjunto. A partir de esta instancia se va seleccionando la `k`-esima en adelante, hasta alcanzar las `s` instancias que componen el conjunto seleccionado.
- Muestreo Doble [SMO96, BFH01]: Se le denomina también Muestreo en dos fases. Consiste en escoger en una primera fase un subconjunto de muestras de tamaño mayor sobre el cuál se obtendrá información adicional sobre los datos. En la segunda fase se obtiene el subconjunto seleccionado final a partir del seleccionado en la fase anterior y empleando la información que se extrajo.
- Muestreo Enlazado [BFH01]: En este caso se aplica un Muestreo o bien aleatorio o estratificado, y se añaden al subconjunto final seleccionado tanto las instancias pertenecientes a ese primer muestreo como todas aquellas que pudieran estar enlazadas o conectadas a ellas.
- Muestreo Inverso [BFH01]: El Muestreo Inverso se caracteriza por repetirse continuamente el proceso de selección del subconjunto solución hasta que este satisface una serie de condiciones específicas.

- Muestreo Progresivo [PJO99]: En los anteriores métodos de Muestreo es necesario fijar el tamaño del subconjunto de muestras a seleccionar. El Muestreo Progresivo comienza con un subconjunto seleccionado de tamaño pequeño. Se aplican múltiples selecciones, evaluándose cada una de ellas. A continuación se aumenta el tamaño del subconjunto a muestrear. El proceso acaba cuando tras incrementar una serie de veces el tamaño del subconjunto no se producen mejoras.

- Selección de Prototipos: Su objetivo es seleccionar un conjunto de muestras tal que mejore la capacidad de predicción de un clasificador basado en la regla del vecino más cercano.

- Combinación de Clasificadores: Recientemente se han desarrollado técnicas para construir conjuntos de clasificadores y combinarlos para clasificar nuevos ejemplos.

  Se ha encontrado que en general son mejores clasificadores, si los clasificadores difieren entre sí, tienen errores menores al 50 % y los errores son independientes entre si.

  Las técnicas desarrolladas de clasificación por votación o conjunta, las podemos dividir en dos grupos: los que cambian la distribución de los ejemplos de entrenamiento (*Boosting*) y los que no (*Bagging*).

  La idea básica es correr un algoritmo de inducción varias veces y combinar los resultados de alguna forma para obtener un mejor resultado final.

  De esta forma se pretende combinar las decisiones tomadas por diferentes clasificadores para confeccionar una decisión conjunta a partir de ellas [TG96], con la que efectuar la selección.

  - *Bagging* (*Boostrap Aggregating*) [Bre96, Qui96, Gra04, EPE04]: *Bagging* genera conjuntos de entrenamiento mediante la selección con reemplazamiento sobre el conjunto inicial. Los conjuntos obtenidos se combinan mediante votación para construir una decisión común.
  - *Boosting* [Fre95, Qui96, KS03]: *Boosting* emplea todas las instancias en cada evaluación, manteniendo un peso para cada instancia del conjunto de entrenamiento que refleja su importancia. El ajuste de estos pesos propicia la aparición de diferentes clasificadores.

    Variantes del modelo *Boosting* son:

  - `PSBoost` (*Prototype Selection Boost*) [NS01]: *Boosting* aplicado a Selección de Prototipos.
  - `AdaBoost` (*Adapting Boosting*) [Sch99, ROM01]: Emplea del mismo modo que *Boosting* el conjunto de clasificadores, sin embargo `Adaboost` lo lleva a cabo secuencialmente.
- *Stacking* (*Stacked Generalization*) [Wol92, Dv04]: Es un modelo de combinación de clasificadores caracterizado por la heterogeneidad de los mismos, frente a la homogeneidad presente en *Bagging* o *Boosting*. Se pretende combinar las clasificaciones realizadas por clasificadores de naturaleza muy diferente.

- Aprendizaje Activo [CAL94, CGJ95, SB02, HR02, BEYL04, ST04]: El proceso de Aprendizaje Activo (*Active Learning*) presenta como diferencia destacable frente a las anteriores técnicas citadas el hecho de que el subconjunto final seleccionado es dinámico. Conforme se van clasificando nuevas instancias, en caso de que estas sean lo suficientemente interesantes, serán agregadas al conjunto de entrenamiento para mejorar así las prestaciones que este proporciona.

1.7. Selección de Prototipos

Los métodos de `SPP` son técnicas de `SII` que pretenden encontrar conjuntos de entrenamiento tales que ofrezcan los mayores porcentajes de clasificación empleando la regla del vecino más cercano (`1-NN`).

La especificación formal del problema es la siguiente: Sean `n` instancias etiquetadas $x_p = (x_{p1}, x_{p2}, \ldots, x_{pm})$, $p = 1, 2, \ldots, n$, con x_p perteneciente a una clase `c` dada y en un espacio `m`-dimensional, donde x_{pi} sería el valor de la `i`-ésima característica de la `p`-ésima muestra. Nuestra tarea consistirá en reducir el conjunto de datos inicial y convertirlo en el menor subconjunto posible caracterizado por permitirnos determinar la clase de una nueva instancia con el mismo o mayor acierto del que conseguíamos con el conjunto original.

El proceso de SPP puede llevar a cabo siguiendo las siguientes estrategias (ver Figura 1.7):

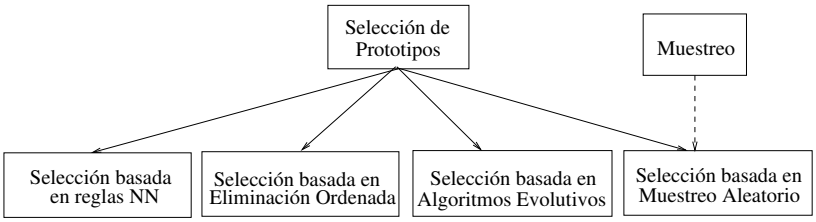

Figura 1.7: Estrategias de selección de prototipos

A continuación se describirá cada una de estas estrategias:

- Selección basada en reglas del vecino más cercano (NN): En este apartado se encuentran todos aquellos métodos que basan sus estrategias de selección en la regla del vecino más cercano. Esta regla se basa en clasificar una nueva instancia como perteneciente a la clase correspondiente a su vecino más cercano.

- Selección basada en eliminación ordenada: En este apartado se encuentran los algoritmos de la familia DROP, de los cuales hemos estudiado los tres primeros dado que los dos últimos no aportan novedades sustanciales. Se basan en la eliminación ordenada de instancias.

- Selección basada en Algoritmos Evolutivos: En este caso la selección se lleva a cabo empleando mecanismos basados en la evolución natural. En la Sección 1.8 y más extensamente en el capítulo siguiente se describe su aplicación.

- Selección basada en Muestreo Aleatorio: En este apartado encontramos métodos de *sampling* que han sido utilizados previamente en problemas de esta naturaleza y que se adecúan a la Selección Prototipos [Ska94].

1.8. Algoritmos Evolutivos y la Extracción de Conocimiento

La razón del éxito de los AAEE es gracias a su habilidad para explotar la información acumulada sobre un espacio de búsqueda del que inicialmente no se conoce nada. Para ello emplean sucesivas búsquedas en subespacios útiles. Esta es su principal característica, particularmente en espacios de búsqueda grandes, complejos y pobremente conocidos, donde las herramientas clásicas son inapropiadas. Ofrecen una aproximación válida a problemas que requieren métodos de búsqueda eficientes y efectivos.

Precisamente éste es el dominio al que las técnicas de extracción de conocimiento deben hacer frente. Se dispone de un conjunto ingente de datos del cual se pretende extraer información útil sin conocimiento a priori. En [Fre02] podemos encontrar un estudio sobre la aplicación de los AAEE a la extracción de conocimiento.

Esta sección se organiza de la siguiente forma: La Subsección 1.8.1 está dedicada a describir los AAEE, ofreciendo las características generales de los mismos [Gol89, Gol02]. En la Subsección 1.8.2 se presenta el empleo de AAEE en el ámbito de la RDD. Para concluir, la Subsección 1.8.3 describe la utilización de los AAEE para el aprendizaje.

1.8.1. Algoritmos Evolutivos

Los AAEE son algoritmos de búsqueda estocásticos que reproducen la evolución biológica natural [BFM97, Gol02]. Todos los AAEE se basan en el concepto de una población de individuos (representando puntos de búsqueda en el espacio de potenciales soluciones del problema), los cuales, empleando operadores probabilísticos de mutación, selección y (en ocasiones) recombinación evolucionan hasta conseguir mayores valores de la función de coste de los individuos. El coste de un individuo refleja el valor de su función objetivo con respecto a una función objetivo particular que se desea optimizar. El operador de mutación introduce innovación dentro de la población al añadir alteraciones en los individuos, mientras que el operador de recombinación realiza un intercambio de información entre diferentes individuos de la población. El operador de selección se encarga de escoger a los mejores individuos para sobrevivir y reproducirse.

Existen diferentes tipos de AAEE: algoritmos genéticos (AAGG), programación genética, estrategias de evolución, y programación evolutiva. Todos los AAEE comparten los mismos conceptos básicos, sin embargo difieren en el mecanismo empleado para codificar las soluciones y los operadores que emplean para producir la siguiente generación.

Los AAEE emplean diferentes parámetros para ajustar su comportamiento. Algunos de ellos son el tamaño de la población, y las tasas de mutación y cruce permitidas. En general, no hay garantía de que el algoritmo evolutivo encuentre la solución óptima, pero una cuidadosa manipulación de los parámetros y de la representación ajustándolos al problema puede incrementar las posibilidades de éxito.

Tenemos diferentes formas de codificar una solución potencial en un cromosoma, del mismo modo que hay diferentes mecanismos de selección, cruce y mutación. Algunas de estas elecciones se ajustan mejor a ciertos problemas particulares que otras y no hay una única mejor elección para todos los problemas. Tradicionalmente, los AAGG emplean cromosomas compuestos de ceros y unos, sin embargo otras codificaciones pueden ser más naturales para algunos problemas y facilitar la búsqueda de buenas soluciones. La programación genética emplea código de programa como soluciones. Las estrategias de evolución y la programación evolutiva utilizan representación en coma flotante, que se adapta mejor al ámbito de optimización de funciones.

Siguiendo el modelo presente en la naturaleza, los AAGG emplean la mutación con probabilidad baja. El cruce es el principal mecanismo para generar nuevos individuos. En su forma más simple, el cruce elige aleatoriamente dos elementos de la población e intercambia dos segmentos del cromosoma entre ellos. Algunas de las nuevas soluciones obtenidas pueden ser mejores que los padres de las que proceden, aunque otras no lo sean. El proceso de selección elimina las peores soluciones, manteniendo las mejores.

La selección se puede llevar a cabo de diferentes formas, pero en todas ellas se persigue el objetivo de preservar buenos individuos, descartando el resto. Los métodos de selección pueden ser suaves o estrictos. En los suaves, se asocia una probabilidad de supervivencia a cada individuo, de forma que aquel que ofrezca la mejor solución será el que presente la mayor probabilidad. En los estrictos, se eligen determinísticamente un número fijo de las mejores soluciones disponibles.

1.8.2. Algoritmos Evolutivos y Reducción de Datos

A continuación presentamos la aplicación de los AAEE a la RDD desde tres perspectivas distintas: Extracción y Selección de Características y SII.

Algoritmos Evolutivos y Extracción de Características

El proceso de extracción de características que son relevantes en un problema depende mucho de los datos. En algunos tipos de datos, las características son relativamente fáciles de identificar. Por ejemplo, en datos de tipo texto, los atributos serían las palabras del texto, o en el caso del análisis de carro de la compra, los atributos serían los productos que han sido adquiridos. En el caso de búsquedas de texto, se eliminan aquellas palabras que no aportan información al texto (por ejemplo, los artículos). En el análisis de carro de la compra, puede que necesitemos convertir las unidades en que los productos han sido comprados para que todos se midan en kilogramos por ejemplo.

Mientras que algunos tipos de datos favorecen la extracción de características, otros la dificultan notablemente. Un ejemplo típico son las imágenes, donde la extracción de características es un reto. Dado su amplio ámbito de utilización, es importante disponer de mecanismos robustos para identificar las características que representan a una imagen. Las imágenes tienden a variar ampliamente, con lo que la naturaleza adaptativa de los AAEE puede ser explotada muy eficazmente.

Dos técnicas tradicionálmente utilizadas para identificar objetos en una imagen son la segmentación, donde la imagen es separada en diferentes regiones basándose en un determinado criterio, y la detección de bordes, donde se identifican los bordes o contornos de una imagen [Wee96].

Algunos autores han explotado el uso de los AAEE en segmentación para tratar espacios de búsqueda grandes y complejos donde existe una información limitada mediante el empleo de la función objetivo. Bhanu y Lee han aplicado AAEE para encontrar de forma adaptativa el conjunto de parámetros de control óptimos para el algoritmo *Phoenix* de segmentación [BL94].

Cagnoni et al. ofrecen otra aproximación al campo de la segmentación mediante el empleo de AAGG para la obtención de imágenes médicas en tres dimensiones [CDPY97].

En el dominio de la detección de fronteras, Bhandarkar et al. propusieron un método basado en minimización de coste, donde el coste tiene en cuenta factores tales como la estructura local del borde, su continuidad y fragmentación [BZP94].

Algoritmos Evolutivos y Selección de Características

Una vez que los atributos representativos del conjunto de datos han sido extraídos, a menudo es de utilidad reducir este número de características. Esto es debido a que en numerosas situaciones no es posible conocer a priori cuales de entre los atributos extraídos serán realmente relevantes. Como añadido, el incluir características irrelevantes no solo aumenta la complejidad en tiempo de muchos algoritmos sino que además eleva el tiempo necesario para seleccionar los atributos.

Siedlecki and Sklansky emplean un algoritmo genético en el que cada individuo de la población está compuesto por ceros y unos, donde un uno indica que el atributo se incluye en el subconjunto y cero que no es así [SS89].

Punch et al. extienden la idea de selección de características binaria mediante la representación de un individuo empleando series de pesos entre cero y diez, ponderando algunos atributos como más importantes que otros [PGP+93].

Yang et al. en [YH98] llevan a cabo la selección de características mediante los `AAGG`.

Ho emplea selección aleatoria de características para crear el conjunto seleccionado [Ho89], siendo extendida por Guerra-Salcedo et al. sustituyendo la selección aleatoria por el algoritmo genético [GSW99].

Casillas et al. en [CCdJH01] presentan un algoritmo genético de selección de características que puede ser integrado en un método de aprendizaje genético multietapa para obtener sistemas de clasificación basados en reglas difusas compuestos por reglas difusas comprensibles con porcentajes de clasificación elevados.

Larrañaga et al. en [IML+01] emplean los `AAGG` y algoritmos de estimación de distribuciones para llevar a cabo la selección de características.

Algoritmos Evolutivos y Selección de Instancias

Del mismo modo que se emplean los `AAEE` para la selección de características, reduciendo el tamaño del conjunto original, se puede optar por disminuir dicho tamaño seleccionando las instancias más representativas.

Como trabajos representativos en este sentido podemos citar el de Kuncheva et al. que emplea un algoritmo genético para editar conjuntos de datos utilizando la regla `k-NN` [Kun95]. En [NI98], Nakashima et al. seleccionan conjuntos de entrenamiento prometedores para el vecino más cercano empleando `AAGG`. Llora et al. en [LG99] emplean los `AAGG` para optimizar un clasificador basado en el

vecino más cercano. En [RB01], Reeves et al. emplean los AAGG para seleccionar conjuntos de entrenamiento con objeto de emplearlos en redes de base radial. LLorá et al. en [LG03] utilizan los AAEE para seleccionar simultáneamente instancias y características con el objetivo de reducir el tamaño del conjunto original. Existen trabajos dirigidos a la selección simultánea de instancias y características mediante el empleo de los AAEE [KJ99, HLL02].

1.8.3. Algoritmos Evolutivos y Aprendizaje

Los AAEE han sido empleados en aprendizaje para la obtención de modelos descriptivos y predictivos. Algunas de sus aplicaciones en éste ámbito son las siguientes:

Algoritmos Evolutivos y Redes Neuronales:

La utilización de los AAGG y redes neuronales han tenido principálmente dos campos de actuación. En el primero de ellos, los AAEE has sido empleados para entrenar o ayudar en el entrenamiento de las redes neuronales. En particular, han sido empleados para ajustar los pesos de la red, buscar los parámetros adecuados de aprendizaje o reducir el tamaño del conjunto de entrenamiento. El segundo campo de actuación ha seguido el camino del diseño de la estructura de la red. Dicha estructura determina enormemente la eficiencia de la red y los problemas que es capaz de resolver [Yao99].

Para el entrenamiento de una red neuronal disponemos de tres mecanismos siguiendo el modelo evolutivo:

- Comenzando con una población aleatoria, utilizar los pesos encontrados por el algoritmo evolutivo en la red neuronal sin refinado adicional [WH89]. Este mecanismo puede ser particularmente útil cuando las funciones de activación de las neuronas no sean diferenciables.
- Emplear retroalimentación u otros métodos para refinar los pesos encontrados por el algoritmo evolutivo [SB95]. La motivación de esta aproximación se basa en que el algoritmo evolutivo puede identificar rápidamente regiones prometedoras en el espacio de búsqueda, pero no realiza una ajuste fino de parámetros de forma rápida.
- Utilizar los AAEE para refinar los resultados obtenidos con un algoritmo de aprendizaje de la red neuronal [KN90]. Aunque los AAEE no refinan la

soluciones de forma rápida, han existido algunos intentos de emplear como población inicial de los AAEE soluciones encontradas con retroalimentación [Man02].

Algoritmos Evolutivos y Extracción de Reglas:

Dado que la programación genética emplea árboles para representar sus soluciones, se adapta perfectamente a la tarea de búsqueda de árboles de decisión. De esta forma, Koza propuso el uso de programación genética en clasificación, donde la función objetivo de cada árbol de decisión valora su capacidad de predicción sobre un conjunto de entrenamiento [Koz92]. Nicolaev y Slavov extienden la función objetivo incluyendo términos relacionados con el tamaño del árbol [NS98].

De Jong et al. desarrollaron GABIL. Se trata de un sistema de aprendizaje de reglas basado en un algoritmo genético. Este sistema aprende y refina reglas de clasificación de conceptos [DSG93].

Giordana et al. en REGAL emplea una implementación paralela para evolucionar conjuntos de poblaciones siguiendo un modelo local de tipo Michigan, y recombinando las poblaciones empleando un algoritmo genético global [GS93]. Una característica interesante de REGAL es el uso de la logica de primer orden para codificar las reglas, lo cual permite identificar relaciones entre reglas más complejas que otros algoritmos basados en codificaciones binarias o numéricas.

GIL, desarrollado por Janikow, presenta la capacidad de aprender múltiples conceptos, permitiendo más de dos etiquetas por clase [Jan93].

Venturini et al. propusieron SIA. Se trata de un algoritmo evolutivo interactivo que permite al usuario evaluar combinaciones de objetivos que describen los datos [VSMdB97]. El objetivo perseguido es encontrar nuevas variables que describan los datos de forma más resumida, y que puedan ser empleadas por algoritmos de clasificación tradicionales.

En [ALF00], Araujo et al. aplican los AAGG paralelos al descubrimiento de reglas en bases de datos de gran tamaño.

Bot emplea programación genética tradicional complementada con un método de selección multiobjetivo para minimizar el tamaño del árbol y el error de clasificación simultáneamente [Bot00].

Riquelme et al. en [RAT00] proponen HIDER para el aprendizaje de reglas en dominios continuos y discretos basándose en los AAEE.

Aguilar presenta COGITO, que consiste en una familia de AAEE cuyo propósito

es obtener un conjunto de reglas de decisión jerárquicas capaz de clasificar un conjunto de datos con la mayor precisión posible en el contexto del aprendizaje supervisado [Agu01].

Bacardit y Garrell proponen un clasificador denominado `GASSIST` que reduce el tamaño de los individuos generalizando soluciones y efectúa un aprendizaje incremental evitando usar todos los ejemplos para llevar a cabo la evaluación [BG02, BG03].

Algoritmos Evolutivos y Redes Bayesianas:

Dentro de la aplicación de los `AAEE` al dominio de las redes bayesianas podemos destacar los siguientes estudios:

En [LL96, LSG$^+$97] se utilizan los `AAGG` para el aprendizaje de la estructura de redes Bayesianas. Zhang en [Zha00] emplea los `AAEE` bayesianos para llevar a cabo aprendizaje y optimización. Larrañaga et al. en [ILS01] efectúan selección de características mediante redes bayesianas comparando `AAGG` y algoritmos secuenciales. Miquélez et al. en [MBL04] estudian los `AAEE` basados en clasificadores bayesianos.

Algoritmos Evolutivos y Sistemas Difusos:

Existen múltiples aplicaciones de los `AAEE` al ámbito de los sistemas difusos. Como referencia podemos citar el libro de Cordón et al. [CHHM01].

Dentro del empleo de los `AAEE` al dominio de los sistemas difusos podemos clasificarlos según su finalidad. Ésta pueda ser:

- Clasificación. Podemos citar trabajos de Sánchez et al. tales como [SC01, SCC01] donde se aplican los `AAEE` para obtener reglas de clasificación difusas.
- Regresión. En [CHS98, CHS99], Cordón et al. efectúan regresión simbólica con algoritmos `GA-P`. Sánchez lleva a cabo regresión simbólica con `AAEE` en [Sán00].

En el siguiente capitulo estudiaremos la aplicación de diferentes `AAEE` a la `SPP`, comparándolos con métodos no evolutivos.

Capítulo 2

Selección Evolutiva de Instancias para la Reducción de Datos

Como pudimos ver en el capítulo anterior, debido a la naturaleza de la información tratada (con ruido, datos incompletos, inconsistentes, etc.) y/o al tamaño del los conjuntos de datos, es útil una etapa en la que se filtran y preprocesan los mismos.

En esta memoria centraremos la atención en la RDD como técnica de preprocesamiento, concretamente en la RDD mediante la SII. Dentro de los mecanismos de SII emplearemos los basados en la SPP.

En la literatura podemos encontrar diferentes propuestas para llevar a cabo la SPP. En [WM00b] se presenta un resumen de las mismas.

Los AAEE han sido aplicados a la SPP con resultados prometedores [Kun95, NI98, RN01, SYCCS02]. La idea es aprovechar la habilidad que presentan como algoritmos de búsqueda para explotar la información acumulada en espacios de búsqueda sin conocimiento a priori.

El objetivo de este capítulo es analizar el uso de los AAEE en la SPP desde una doble vertiente: clasificación basada en el vecino más cercano y selección

de conjuntos de entrenamiento para generar modelos predictivos mediante C4.5 [Qui93].

El capítulo aparece organizado en las siguientes secciones. En la Sección 2.1, se describen las dos vertientes seguidas en la evaluación de los algoritmos de SPP. En la Sección 2.2, se presentan las diferentes técnicas de SPP no evolutivas utilizadas en esta memoria. La Sección 2.3 está dedicada a describir los AAEE estudiados y su aplicación en SPP. La Sección 2.4 presenta la metodología de experimentación seguida. En la Sección 2.5, se muestran los resultados y su análisis. La Sección 2.6 analiza el comportamiento de los AAEE en SPP. La Sección 2.7 está dedicada a las conclusiones finales extraídas del capítulo.

2.1. Estrategias Seguidas en Selección de Instancias: Clasificación basada en Prototipos y Selección de Conjuntos de Entrenamiento

La SPP se puede llevar a cabo siguiendo dos objetivos diferentes: considerando mejorar la capacidad de predicción del clasificador basado en la regla del vecino más cercano (objetivo al que llamaremos clasificación), o bien la selección de conjuntos de entrenamiento a partir de los que obtener modelos descriptivos o predictivos posteriormente. A continuación describimos cada una de dichas estrategias:

- Clasificación basada en la SPP mediante la técnica del vecino más cercano: En este caso se desea mejorar el comportamiento del clasificador basado en el vecino más cercano mediante la selección de las muestras de entrenamiento con mayor representatividad y capacidad de generalización. La Figura 2.1 muestra el proceso:

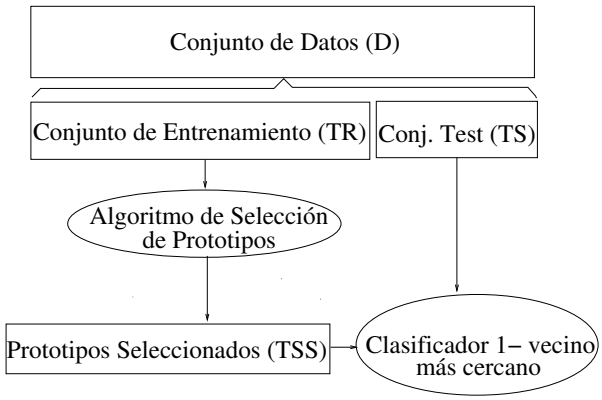

Figura 2.1: SPP aplicada a clasificación

El conjunto inicial (`D`) se divide en dos, uno sobre el que se llevará a cabo la selección (`TR`) y otro que será empleado para validarla (`TS`). Sobre el conjunto `TR` se aplica el algoritmo de selección para obtener el conjunto de prototipos que se emplearan como conjunto de entrenamiento en el clasificador 1-vecino más cercano, empleando como conjunto de test a `TS`.

- Selección de Conjuntos de Entrenamiento: El objetivo perseguido es obtener aquel conjunto de instancias tal (al que denominamos `TSS`), que independientemente del algoritmo de aprendizaje que se le aplique para obtener el modelo, mantenga su representatividad y capacidad de generalización (ver Figura 2.2).

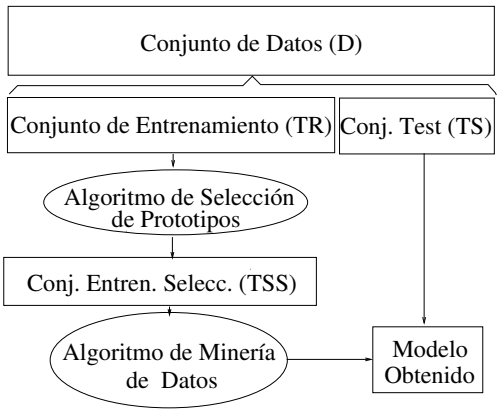

Figura 2.2: SPP aplicada a selección de conjuntos de entrenamiento

Del mismo modo que en el caso anterior, el conjunto inicial (`D`) se divide en dos, `TR` y `TS`. Sobre `TR` se aplica el algoritmo de selección para obtener `TSS` como conjunto de entrenamiento seleccionado. Este conjunto `TSS` se emplea como entrada para el algoritmo de minería de datos (en este estudio se emplea `C4.5`). `C4.5` generará un modelo a partir del conjunto `TSS` de entrada que será validado empleando `TS`.

2.2. Técnicas No Evolutivas de Selección de Instancias

En la Sección 1.7 aparecen clasificadas las diferentes vías de `SPP` (ver Figura 2.3). A continuación se describirán los algoritmos de `SPP` siguiendo esta clasificación[1]:

[1]NOTA: Se han publicado algunas nuevas propuestas de selección de prototipos que no han sido recogidas en el presente capítulo debido a la reciente aparición de las mismas. El presente capítulo corresponde al estudio publicado en [CHL03b]

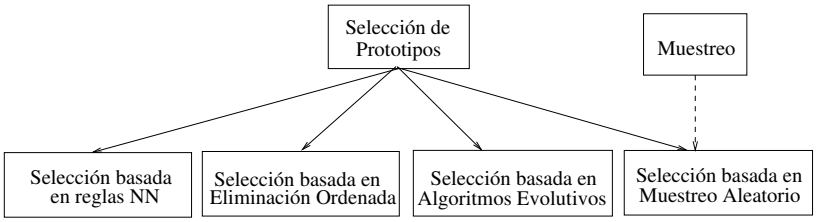

Figura 2.3: Estrategias de selección de prototipos

- Selección basada en la regla del vecino más cercano (reglas `NN`):
  - *Condensed Nearest Neighbour* (`CNN`) [Har68]: Este algoritmo busca un subconjunto `TSS` del conjunto de entrenamiento `TR` tal que cada miembro de `TR` está más cercano a un miembro de `TSS` de la misma clase que a un miembro de `TSS` de diferente clase.
  - *Edited Nearest Neighbour* (`ENN`) [Wil72a]: El algoritmo comienza igualando `TSS` a `TR` para a continuación ir eliminando de `TSS` aquella instancia tal que no se clasifique correctamente según sus `k` vecinos más cercanos.
  - *Repeated Edited Nearest Neighbour* (`RENN`) [Wil72a]: Consiste en una aplicación repetida de `ENN` hasta que todas las instancias presentes estén rodeadas de muestras de su misma clase.
  - *Reduced Nearest Neighbour* (`RNN`) [Gat72]: Comienza de nuevo igualando `TSS` a `TR`. A continuación trata de eliminar una muestra siempre y cuando dicha eliminación no produzca que el resto de instancias en `TR` se clasifiquen incorrectamente por las instancias restantes en `TSS`.
  - *Variable Similarity Metric* (`VSM`) [Low95]: En esta técnica se iguala `TSS` a `TR`, y a continuación se van eliminando instancias de `TSS`. Una muestra x_p de `TSS` es eliminada si la mayoría (más de un 60 %) de sus `k` vecinos más cercanos son de la misma clase que x_p, o bien, si la mayoría de sus `k` vecinos más cercanos son de una clase diferente, con lo que podemos concluir que x_p es ruido.
  - `Multiedit` [DK82]: Consiste en una modificación del algoritmo `ENN` que garantiza la independencia estadística entre los prototipos retenidos en el conjunto resultante. Se trata de realizar una partición del conjunto inicial de prototipos para obtener `m` muestras significativas.

Después, se procede a aplicar el ENN sobre cada una de las particiones obtenidas. Finalmente, el conjunto editado estará formado por los prototipos retenidos en las m muestras.

- *Model Class Selection* (MCS) [Bro93]: Se basa principalmente en mantener un registro de las veces en las que una instancia se encuentra dentro de los k vecinos más cercanos de otra instancia, dándose el caso de que coincidan ambas clases.
- Shrink [KA87]: En este caso igualamos TSS a TR, y se van eliminando aquellas instancias de TSS que son clasificadas correctamente por el subconjunto restante.
- *Instance Based 2* (IB2) [KA87]: Comenzamos con el subconjunto TSS vacío. A continuación, cada muestra de TR se va añadiendo en TSS si dicha muestra no es clasificada correctamente por las restantes muestras presentes en TSS.
- *Instance Based 3* (IB3) [AKA91]: Comenzamos con el subconjunto TSS vacío. Este algoritmo va añadiendo muestras de TR en TSS, de tal forma que solo añade aquellas muestras mal clasificadas consideradas como aceptables.
- *Iterative Case Filtering* (ICF) [BM02]: Dicho algoritmo pretende mantener aquellas muestras del conjunto TR que clasifican correctamente una mayor cantidad de ejemplos. Para ellos utiliza los conceptos de alcanzabilidad y cobertura de unas muestras sobre otras.

▪ Selección basada en eliminación ordenada. A destacar:

- *Decremental Reduction Optimization Procedure 1* (DROP1) [WM97]: Este algoritmo es similar al RNN, diferenciándose en que se comprueba si una muestra se puede quitar o no en el conjunto TSS en vez de hacerlo en el TR.
- *Decremental Reduction Optimization Procedure 2* (DROP2) [WM97]: La heurística que se sigue es eliminar una muestra x_p si al menos la mayoría de sus asociados (aquellas muestras que tienen a x_p como muestra más cercana) en TR podrían clasificarse correctamente sin esa muestra. DROP2 cambia el orden de eliminación de instancias de tal forma que se comienza dicha eliminación por aquellas que están más distantes de su enemigo (muestra de clase distinta) más cercano.

  - *Decremental Reduction Optimization Procedure 3* (DROP3) [WM97]: La principal aportación de DROP3 frente a DROP2 consiste en aplicar un proceso de filtrado de ruido antes de la eliminación ordenada de instancias. La regla de filtrado seguida consiste en desestimar cualquier instancia mal clasificada por sus k vecinos más cercanos.

- Selección basada en Algoritmos Evolutivos. Dentro de esta categoría situaríamos nuestra propuesta. En la Sección 2.3 se describe en profundidad las características generales de los AAEE estudiados, así como las peculiaridades de cada uno ellos.

- Selección basada en Muestreo Aleatorio. Las dos técnicas estudiadas son:

  - *Random Mutation Hill Climbing* (Rmhc) [Ska94]: Se selecciona un subconjunto TSS de instancias pertenecientes a TR de un tamaño previamente determinado s. En cada iteración el algoritmo quita una de las muestras presentes en TSS y añade otra del conjunto TR-TSS, de tal forma que si dicho cambio mejora las prestaciones del subconjunto se mantiene el cambio, en caso contrario se deshace.
  - *Edited Nearest Neighbour by Random Selection* (Ennrs) [WM00b]: Se trata de un algoritmo semejante al anterior solo que en cada iteración se reemplazan las s muestras del conjunto seleccionado TSS por otras pertenecientes a TR, quedándonos siempre con el conjunto TSS que ofrezca mejores resultados. Este mecanismo es especialmente útil para problemas de elevada dificultad donde podría aparecer una elevada cantidad de ruido, limitadas muestras de entrenamiento o datos con demasiados atributos.

2.3. Algoritmos Evolutivos Aplicados a Selección de Instancias

Los AAEE han sido propuestos por diferentes autores para abordar el problema de SPP [Kun95, LG99, RN01, SYCCS02].

Esta Sección esta organizada de la siguiente forma. En la Subsección 2.3.1 vamos a describir en primer lugar cada uno de los AAEE empleados en nuestro

estudio. Posteriormente, en la Subsección 2.3.2 se muestra el esquema de representación de soluciones empleado para poder codificar el problema de SPP. Finalmente se presenta la función objetivo (*fitness*) que se desea optimizar en la Subsección 2.3.3.

2.3.1. Algoritmos Evolutivos Utilizados

Los AAEE evaluados son los siguientes:

- Algoritmo Genético Generacional (AGG) [Hol75, Gol02]: Los AAGG son algoritmos de búsqueda de propósito general cuyos principios están basados en la evolución natural donde cada individuo representa una solución del problema. La idea básica consiste en mantener una población de cromosomas, que representan soluciones candidatas, que evolucionan a través de sucesivas iteraciones (generaciones) empleando un proceso de competición y variación controlada. Cada cromosoma en la población tiene un coste asociado que determinará los cromosomas que se van a utilizar para formar los nuevos, en un proceso competitivo llamado selección. Los nuevos cromosomas son creados empleando operadores genéticos tales como el cruce y la mutación.

  Aunque hay muchas posibles variantes del algoritmo genético básico, el modelo clásico es el AGG, que consiste en tres operaciones:

  1. *Evaluación del coste de los individuos.*
  2. *Formación de una población intermedia empleando un mecanismo de selección.*
  3. *Recombinación a través de cruces y mutaciones.*

  A continuación se muestra la estructura de un algoritmo genético básico. $P(t)$ indica la población en la generación t.

```
Empezar;
    t = 0;
    Inicializar P(t);
    Evaluar P(t);
    Mientras (no condicion de terminacion) hacer
    Empezar
        t = t + 1;
        Seleccionar P(t) a partir de P(t − 1);
        Recombinar P(t);
        Evaluar P(t);
    Fin;
Fin;
```

El mecanismo de selección produce una nueva población, $P(t)$, con copias de cromosomas de $P(t-1)$. El número de copias recibidas de cada cromosoma depende del valor alcanzo en la función objetivo; cromosomas con mayor valor suelen tener una mayor oportunidad de contribuir con copias en $P(t)$. Entonces, el operador de cruce y mutación se aplica sobre $P(t)$.

El cruce toma dos individuos llamados padres y produce dos nuevos individuos llamados descendencia mediante el intercambio de partes de los padres. En su forma más simple el operador intercambia subcadenas a partir de un punto de cruce seleccionado de forma aleatoria. El operador de cruce normalmente no se aplica a todos los pares de cromosomas en la nueva población. Se suele aplicar a un porcentaje de la población. Al porcentaje se le llama probabilidad de cruce.

La mutación se emplea para prevenir la pérdida prematura de diversidad en la población. Para ello se muestrean aleatoriamente nuevos puntos en el espacio de búsqueda. La probabilidad de mutación se mantiene baja, para evitar que el proceso degenere en una búsqueda aleatoria. En cromosomas binarios, la mutación se aplica cambiando el estado de uno o más bits aleatorios dentro del cromosoma, empleando para decidir cuando se muta un parámetro que representa la probabilidad de mutación.

La finalización del proceso se puede determinar controlando cuando se alcanza un número máximo de generaciones o utilizando algún criterio de terminación cuando se encuentre una solución aceptable.

- Algoritmo Genético Estacionario (AGE) [Whi89]: En el AGE normalmente se producen uno o dos descendientes en cada generación. Los padres son seleccionados para producir descendencia y se define una estrategia de eliminación o reemplazo para decidir que miembros de la población serán sustituidos por los nuevos descendientes. La idea básica del AGE es la siguiente:

  1. *Seleccionar dos padres de la población* P.
  2. *Crear una descendencia empleando cruce y mutación.*
  3. *Evaluar la descendencia utilizando la función de coste.*
  4. *Seleccionar los individuos en* P *que serán sustituidos por la descendencia.*
  5. *Decidir si ese individuo será reemplazado.*

  En el cuarto paso se decide la estrategia de reemplazo, que puede ser por ejemplo reemplazar el peor de la población (el que tiene peor valor según la función de coste), el más antiguo, o uno aleatorio. En el paso 5 se decide la condición de reemplazo, que podría ser por ejemplo, reemplazar si el nuevo individuo es mejor o un reemplazo sin condiciones. La combinación ampliamente utilizada es reemplazar el peor individuo sólo si el nuevo individuo es mejor (esta es la que nosotros hemos empleado).

- CHC [Esh91]: Este algoritmo es un método ya clásico en la literatura especializada de algoritmos genéticos, siendo una de las primeras propuestas en introducir mecanismos de diversidad para alcanzar un buen equilibrio entre explotación y exploración en el proceso de búsqueda. En cada generación el algoritmo CHC utiliza una población de padres de tamaño N para generar una población intermedia de N individuos, que se emparejan aleatoriamente y se emplean para generar los N potenciales descendientes. En este momento comienza una competición por la supervivencia, donde los N mejores cromosomas de entre las poblaciones de padres e hijos son seleccionados para conformar la siguiente generación.

  El algoritmo CHC implementa un mecanismo de recombinación heterogéneo que utiliza HUX, un operador de recombinación especial. HUX intercambia la mitad de los bits en los que difieren los padres, donde la posición del bit a intercambiar se determina de forma aleatoria. CHC introduce un método

para prevenir el incesto. Antes de aplicar `HUX` entre dos padres, se calcula la distancia de *Hamming* (número de bits en la misma posición del cromosoma en los que difieren) entre ellos. Solo se cruzan aquellos padres que presenten una distancia de *Hamming* superior a un umbral. El umbral inicialmente comienza valiendo `L`/4, donde `L` es la longitud de los cromosomas. En el momento en el que no se inserten descendientes en la nueva población el umbral se reduce en 1.

No se aplica ningún proceso de mutación durante la fase de recombinación. En lugar de eso, cuando la población converge o el proceso de búsqueda deja de progresar adecuadamente (el umbral de cruce del que hablábamos antes llega a 0 y no se generan nuevos descendientes), la población se reinicializa para introducir nueva diversidad en la búsqueda. El cromosoma que representa la mejor solución encontrada hasta ese momento se utiliza como patrón para generar la nueva población. Esto se lleva a cabo generando cada nuevo cromosoma alterando en un 35 % los bits del cromosoma patrón, hasta generar `N`-1 cromosomas de la nueva población. Se generan `N`-1 dado que el cromosoma patrón se incorpora a la nueva población.

- *Population-Based Incremental Learning* (`PBIL`) [Bal94]: `PBIL` se puede considerar como el modelo básico e inicial de los algoritmos de evolución basados en estimaciones de distribuciones (modelos `EDA`). En [LL01] se puede encontrar una descripción completa de los mismos.

  `PBIL` es una combinación de algoritmo genético y aprendizaje competitivo. Fue diseñado para búsquedas en espacios binarios. El algoritmo `PBIL` mantiene explícitamente estadísticas sobre el espacio de búsqueda para decidir cuál es el siguiente conjunto a muestrear.

  El algoritmo crea un vector de probabilidades, v_p, que se emplea para generar soluciones de mayor calidad. Esto se llevaría a cabo de la siguiente forma: una solución se codifica como una cadena compuesta por 0 y 1 en cada posición, un posible vector v_p podría ser 0.01, 0.99, 0.01, 0.99, etc. En un primer momento los valores de v_p están inicializados a 0.5. Al principio, al generar nuevas muestras empleando v_p la probabilidad de generar un 1 o un 0 para una solución es la misma. Conforme la búsqueda progresa, los valores de v_p se van desplazando gradualmente para representar los vectores con soluciones de más alta calidad siguiendo el siguiente proceso:

1. *Se genera un número de vectores solución (N_{pob}) basándose en las probabilidades especificadas en v_p.*
2. *Los valores de v_p se desplazan hacia el vector solución generado con mejor coste asociado, S_{mejor}.* `LR` *es la tasa de aprendizaje, que especifica la rapidez con la que se acerca v_p a la mejor solución.*

$$v_p[i] = v_p[i] \cdot (1 - LR) + S_{mejor}[i] \cdot LR \tag{2.1}$$

3. *Los valores de v_p se alejan así mismo del vector solución generado con peor coste, S_{peor}.* `NegatLR` *es la tasa de aprendizaje negativo, que especifica la rapidez con la que se aleja v_p de la peor solución.*

$$Si\ S_{mejor}[i] <> S_{peor}[i]\ Entonces$$
$$v_p[i] = v_p[i] \cdot (1 - NegatLR) + S_{mejor}[i] \cdot NegatLR$$

4. *Tras actualizar el vector de probabilidades, se generan nuevos vectores solución empleando el nuevo vector de probabilidades v_p que se acaba de obtener y el ciclo continúa.*

`PBIL` aplica mutación sobre v_p con el mismo propósito que la mutación en un algoritmo genético: evitar la convergencia prematura. Se utiliza una mutación con una pequeña probabilidad P_m sobre v_p, en una dirección aleatoria `Mut_Desplaz` (hacia 0 o hacia 1).

2.3.2. Esquema de Representación

Partimos de un conjunto de datos al que denominaremos `TR` compuesto por `n` instancias. De esta forma, el espacio de búsqueda estará constituido por todos los subconjuntos de `TR`. Cada cromosoma es uno de esos subconjuntos. La solución estará representada empleando un cromosoma binario con `n` genes, donde cada gen puede presentar dos posibles estados: 1 ó 0, indicando pertenencia o no al conjunto seleccionado respectivamente. La Figura 2.4 muestra un ejemplo de la representación empleada.

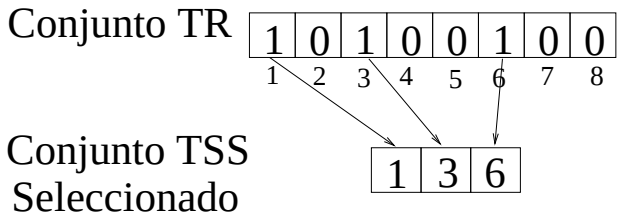

Figura 2.4: Representación de las soluciones candidatas en la selección de prototipos

2.3.3. Función Objetivo

Sea `TSS` un subconjunto de instancias de `TR` codificadas en un cromosoma para ser evaluado. Definiremos la función de evaluación como combinación de dos valores: el porcentaje de clasificación asociado a `TSS` (`porc_clas`) y el porcentaje de reducción conseguido en `TSS` con respecto a `TR` (`porc_red`):

$$\texttt{F\_Eval}(\texttt{TSS}) = \alpha \cdot \texttt{porc\_clas} + (1 - \alpha) \cdot \texttt{porc\_red}. \tag{2.2}$$

Se emplea el clasificador 1-vecino más cercano para calcular el porcentaje de clasificación (`porc_clas`). Dicho porcentaje representa el porcentaje de muestras clasificadas correctamente de `TR` empleando tan solo instancias de `TSS` para encontrar el vecino más cercano. Para cada objeto y en `TR`, se busca su vecino más cercano entre aquellos pertenecientes a $\texttt{TSS} \setminus \{y\}$.

El porcentaje de reducción (`porc_red`) se obtiene de la siguiente forma:

$$\texttt{porc\_red} = 100 \cdot (|\texttt{TR}| - |\texttt{TSS}|)/|\texttt{TR}|. \tag{2.3}$$

El objetivo de los `AAEE` es maximizar la función de evaluación definida. Para ello deben maximizar la combinación del porcentaje de clasificación y el de reducción de acuerdo con el parámetro α.

2.4. Metodología de Experimentación

La experimentación se lleva a cabo empleando dos tipos de conjuntos de datos según su tamaño: pequeños y medianos. Así, disponemos de diez conjuntos de datos de tamaño pequeño y tres de tamaño mediano. En su evaluación empleamos validación cruzada de tamaño 10 sobre ellos. Los parámetros empleados en los algoritmos que los necesitan van a estar determinados, o bien por la bibliografía (caso de `Ib3`) o bien por resultados empíricos que hemos obtenido (caso del algoritmo `Rmhc` por ejemplo).

En el estudio realizado se han introducido como referencias los algoritmos 1-vecino más cercano (`1-NN`: *1-Nearest Neighbour*) para clasificación y el `C4.5` para selección de conjuntos de entrenamiento. De este modo podemos estudiar el comportamiento del resto de algoritmos tras efectuar la reducción sobre el conjunto de datos.

La Subsección 2.4.1 describe los conjuntos de datos empleados, junto con sus características. En la Subsección 2.4.2 se presenta la validación cruzada seguida en los experimentos y los parámetros de cada algoritmo que se han utilizado.

2.4.1. Conjuntos de Datos

Todos los conjuntos de datos han sido obtenidos del Repositorio de la UCI [MM96].

Los conjuntos de datos empleados de tamaño pequeño aparecen reflejados en la Tabla 2.1:

Tabla 2.1: Conjuntos de Datos de Tamaño Pequeño

Conj. Datos	Núm. Instancias	Núm. Atributos	Núm. Clases
Cleveland	303	13	5
Glass	294	9	6
Iris	150	4	3
Led24Digit	200	24	10
Led7Digit	500	7	10
Lymphography	148	18	4
Monk	432	6	2
Pima	768	8	2
Wine	178	13	3
Wisconsin	699	10	2

En cada uno de ellos el objetivo es el siguiente:

- *Cleveland*: Contiene 76 atributos, sin embargo todos los experimentos publicados emplean tan solo un subconjunto que contiene 13 de ellos. El objetivo en esta base de datos consiste en predecir la existencia de una enfermedad cardíaca en un paciente.

- *Glass*: Consiste en clasificar diferentes tipos de cristal. Esta motivado por investigaciones criminológicas. En la escena del crimen, el cristal presente puede ser utilizado como evidencia si es correctamente clasificado.

- *Iris*: El conjunto de datos contiene clases de 50 instancias cada una, donde cada clase se refiere a un tipo de planta iris. Una clase es linealmente separable de las otras dos; las dos últimas no son linealmente separables una de otra.

- *LED24Dig*: Este dominio contiene 24 atributos booleanos correspondientes a 10 conceptos, que son los 10 dígitos decimales. Un modelo de display de

tipo LED contiene 24 diodos emisores de luz – de aquí la razón de los 24 atributos. El problema no sería difícil de no ser por la introducción de ruido.

- *LED7Dig*: Este dominio, a semejanza del anterior, contiene 7 atributos Booleanos asociados a 10 conceptos, que son los 10 dígitos decimales. Un modelo de display de tipo LED contiene 7 diodos emisores de luz.
- *Lymphography*: Esta base de datos ha sido obtenida del *University Medical Centre, Institute of Oncology, Ljubljana*, Yugoslavia.
- *Monk*: Este problema ha sido la base de la primera comparativa internacional de algoritmos de aprendizaje. Los resultados de dicha comparativa aparecen reunidos en *"The MONK's Problems"*.
- *Pima*: En este problema se pretende llevar a cabo el diagnóstico de una variable binaria para deducir si el paciente muestra signos de diabetes según el criterio de la organización mundial de la salud.
- *Wine*: Estos datos son producto de análisis químicos llevados a cabo en vinos de la misma region de Italia, pero derivados de tres cultivos diferentes.
- *Wisconsin*: Esta base de datos de cáncer de pecho ha sido obtenida de la *University of Wisconsin Hospitals, Madison* del Dr. William H. Wolberg.

Los conjuntos de tamaño mediano evaluados son los que aparecen en la Tabla 2.2:

Tabla 2.2: Conjuntos de Datos de Tamaño Mediano

Conj. Datos	Núm. Instancias	Núm. Atributos	Núm. Clases
Pen-Based Recognition	10992	16	10
Satimage	6435	36	6
Thyroid	7200	21	3

A continuación se presenta una breve descripción de los mismos:

- *Pen-Based Recognition*: Consiste en una base de datos creada al tomar 250 muestras de 44 escritores. Para ello se empleó una tarjeta digitalizadora

sensible a la presión. A los participantes se les pidió escribir 250 dígitos en orden aleatorio dentro de cajas de resolución de 500 por 500. Los datos recogidos de la tarjeta digitalizadora son valores enteros entre 0 y 500.

- *Satimage*: La base de datos consiste en valores multiespectrales de píxeles en una vecindad de 3x3 de una imagen tomada por satélite y su clasificación asociada con el píxel central en cada conjunto de vecinos. En el conjunto de muestras, la clase de un pixel viene codificada como un número.

- *Thyroid*: El problema consiste en determinar cuando un paciente padece hipotiroidismo. Podemos encontrar tres clases: normal (no hipotiroidismo), hiperfunción y funcionamiento anormal.

2.4.2. Validación Cruzada y Parámetros de los Algoritmos

Los conjuntos considerados se dividen empleando un procedimiento de validación cruzada de tamaño 10. Cada conjunto D es dividido en 10 subconjuntos disjuntos de igual tamaño, $D_1, ..., D_{10}$, manteniendo la distribución de clases. Para la validación `i` vamos a considerar los siguientes conjuntos:

$$TR_i = \bigcup_{1 \leq j \leq 10, j \neq i} D_j \tag{2.4}$$

$$TS_i = D_i \tag{2.5}$$

Las expresiones 2.4 y 2.5 representan el modo en el cual se definen el conjunto de entrenamiento TR_i y test TS_i para la validación `i`.

Se han efectuado 10 ejecuciones de cada algoritmo determinístico, una para cada validación `i` considerada. Los resultados que aparecerán en las tablas posteriores estarán compuestos por la media de esas 10 ejecuciones.

En aquellos algoritmos con componente aleatorio para cada validación `i` se han efectuado 3 ejecuciones. Las 3 ejecuciones por los 10 subconjuntos presentes hacen un total de 30 ejecuciones. Los resultados de estos últimos algoritmos serán la media de esas 30 ejecuciones.

Los parámetros empleados por los algoritmos están reflejados en la Tabla 2.3:

Tabla 2.3: Parámetros de los Algoritmos

Algoritmo	Parámetros
Ib3	Aceptabilidad=0.9, Eliminación=0.7
Rmhc	m=90 % , Eval=10000
Ennrs	m=90 % , Eval=10000
AGG	P_m=0.001, P_c=0.6, Pob=50, Eval=10000, α=0.5
AGE	P_m=0.001, P_c=1, Pob=50, Eval=10000, α=0.5
CHC	Pob=50 , Eval=10000, α=0.5
PBIL	LR=0.1, $Mut_{desplaz}$=0.05, P_m= 0.02,
	Pob=50, $NegatLR$=0.075, Eval=10000, α=0.5

2.5. Estudio Experimental

En esta sección presentamos los resultados de los experimentos efectuados y el análisis de los mismos. En la Subsección 2.5.1 se describen las tablas que contendrán los resultados. La Subsección 2.5.2 muestra los resultados y su análisis en la SPP aplicada a clasificación. En la Subsección 2.5.3 se encuentra los resultados y el análisis de los algoritmos de SPP empleados para la selección de conjuntos de entrenamiento.

2.5.1. Estructura de las Tablas de Resultados

Con las 10 ó 30 ejecuciones por algoritmo confeccionamos la tabla de resultados para compararlos, donde se presentará el valor medio de dichas ejecuciones.

Dada la gran cantidad de tablas e información de la que disponemos, para facilitar el proceso de análisis en cada subsección de resultados incluiremos una tabla con los resultados medios conseguidos por los algoritmos en los conjuntos de datos, pequeños o medianos, siguiendo el formato anteriormente indicado. Esta tabla será empleada como base para el posterior análisis del comportamiento de los algoritmos. Las tablas van a presentar la siguiente estructura:

- La primera columna contiene el nombre de cada algoritmo.
- La segunda columna presenta los tiempos de ejecución medios en segundos.
- La tercera está destinada a los porcentajes de reducción ofrecidos por cada algoritmo.
- La cuarta columna muestra el porcentaje de acierto obtenido al aplicar el clasificador `1-NN` (para clasificación) o `C4.5` (en selección de conjuntos de entrenamiento) sobre el conjunto de datos TSS_i para clasificar TR_i (porcentaje de acierto en entrenamiento).
- La ultima columna ofrece el porcentaje de acierto obtenido por `1-NN` (para clasificación) o `C4.5` (en selección de conjuntos de entrenamiento) sobre TS_i utilizando TSS_i como conjunto de entrenamiento (porcentaje de acierto en test).

Estudiando tan solo la tabla de resultados medios surgen dos inconvenientes:

- Al mostrarse únicamente el valor medio se oculta información importante, dado que es posible que un algoritmo se comporte excepcionálmente bien en un conjunto de datos sin hacer lo mismo en el resto. El hecho de que la media de un algoritmo sea mejor que el resto no significa necesariamente que sea mejor que el resto de métodos.
- El porcentaje de clasificación medio evaluado sobre diferentes conjuntos de datos no tiene suficiente significado por sí solo. Este hecho es debido a que cada problema puede presentar diferentes grados de dificultad. Por ejemplo, en un problema donde el porcentaje de clasificación referencia esté en un 60 %, y el algoritmo obtenga un 80 % puede decirse que es un muy buen resultado. En el caso en que la referencia esté en un 90 % y el algoritmo proporcione un 80 % podríamos decir que es un mal resultado.

Para aportar más información que pueda paliar estos inconvenientes hemos incluido un segundo tipo de tabla de resultados. Esta tabla va a mostrar la ordenación de los algoritmos según la calidad de sus resultados, junto con el número de veces que obtienen el mejor resultado. De esta forma, al tener información sobre el algoritmo que obtiene un mayor número de veces el mejor resultado en los conjuntos de datos se elimina el primer inconveniente previamente citado.

Para confeccionar la ordenación media el proceso seguido es el siguiente: Para cada conjunto de datos se ordenan lo algoritmos de mejor a peor resultado según un determinado objetivo. En el caso del porcentaje de acierto, se ordenarían situando en primera posición aquel con el mejor porcentaje y en el último al que presente el peor. Se calculará la ordenación según un objetivo dado (acierto, reducción o combinado) para todos los conjuntos de datos. El ranking medio de un algoritmo es la media de las posiciones obtenidas en los rankings calculados para cada conjunto de datos. Este ranking clarifica los resultados obtenidos evitando el segundo inconveniente anteriormente citado.

Para construir esta tabla se han empleado los resultados proporcionados por cada algoritmo en los conjuntos de datos evaluados (pequeños o medianos). Dado que la reducción del conjunto de datos es objetivo clave en este estudio, hemos seleccionado para confeccionar esta segunda tabla a aquellos algoritmos con una tasa de reducción superior al 70 %. Dichos algoritmos aparecerán marcados en la tabla de resultados medios mediante un asterisco (*) junto a su nombre.

En esta tabla encontraremos tres clasificaciones según un determinado objetivo: reducción, acierto o ambos valorados al 50 %. Para generar la ordenación según el porcentaje de reducción, por ejemplo, se ordenan de forma descendente los algoritmos seleccionados según este porcentaje. Al primer algoritmo en este orden se le asigna un valor de 1, al segundo un 2, siguiendo este proceso de manera sucesiva. La clasificación media de un algoritmo se presenta como una medida que varía entre 1 y el número de algoritmos comparados. Será un valor medio dado que dependerá de su posición en la ordenación en los 10 conjuntos de tamaño pequeño o en los 3 de tamaño mediano.

La estructura que presentará este tipo de tabla es la siguiente:

- La primera columna muestra la ordenación de los algoritmos considerando como objetivo el porcentaje de reducción. La columna se subdivide en 3 subcolumnas. La primera (Nom), contiene el nombre del algoritmo, la segunda (Ord) muestra su ordenación media, y la tercera y última (Mej), representa el número de veces que el algoritmo ha sido el mejor según ese objetivo.
- La segunda columna ofrece el mismo tipo de información que la primera, considerando el porcentaje de acierto en test como único objetivo.
- La tercera presenta la misma información que las dos anteriores, considerando en este caso como objetivo la combinación al 50 % de reducción y

acierto en test.

2.5.2. Resultados y Análisis en Clasificación

En esta sección incluimos los resultados en SPP aplicada a clasificación, estudiando el comportamiento de las diferentes técnicas al aplicarlas sobre conjuntos de diferente tamaño.

2.5.2.1. Resultados en Clasificación para Conjuntos de Tamaño Pequeño

En la Sección 2.A ofrecemos los resultados para cada uno de los conjuntos de tamaño pequeño evaluados. La Tabla 2.4 muestra los resultados medios y la Tabla 2.5 ofrece la ordenación de los algoritmos por objetivo (reducción, acierto, o reducción-acierto):

Tabla 2.4: Resultados Medios de Selección de Prototipos en Clasificación en Conjuntos de Tamaño Pequeño

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0.07		68.44	65.73
Cnn (*)	0.01	71.98	49.11	52.18
Drop1(*)	0.23	86.97	68.10	62.74
Drop2	0.20	55.19	67.23	67.16
Drop3(*)	0.15	78.56	86.62	63.80
Enn	0.10	35.07	72.67	66.37
Ib2 (*)	0.03	77.80	47.75	51.36
Icf (*)	0.29	75.81	66.08	62.01
Mcs	0.09	16.90	74.56	67.08
Multied	0.24	54.70	61.33	61.06
Renn	0.25	37.43	72.30	65.97
Rnn (*)	1.89	90.38	68.17	65.51
Shrink	0.08	30.41	72.05	66.64
Vsm (*)	0.01	74.56	59.89	58.87
Ib3	0.06	65.71	60.42	64.67
Rmhc(*)	54.37	90.16	57.02	58.17
Ennrs (*)	69.95	90.16	69.52	64.13
AGG (*)	70.8	87.72	77.12	67.54
AGE (*)	68.6	90.50	77.89	67.65
CHC (*)	20.48	96.05	75.86	64.37
PBIL (*)	43.2	93.79	72.49	64.63

Tabla 2.5: Ordenación de los Algoritmos por Objetivo en Clasificación para Conjuntos de Tamaño Pequeño

% Reducción			% Test 1-NN			%Test 1-NN - %Reducción		
Nom	Ord	Mej	Nom	Ord	Mej	Nom	Ord	Mej
CHC	1.3	7	AGE	3.5	2	PBIL	2.7	4
PBIL	2.9	1	AGG	4.2	0	AGE	3.4	1
AGE	5.1	0	PBIL	4.3	5	CHC	3.5	4
Rnn	5.8	0	Ennrs	5.4	0	Rnn	4.6	0
Rmhc	5.8	1	Rnn	5.6	1	Ennsr	5.3	1
Drop1	6.6	1	CHC	6.6	1	AGG	5.5	0
Ennrs	6.8	0	Drop1	7.2	1	Rmhc	7.5	0
Ib2	7.3	0	Drop3	7.2	0	Drop1	7.6	0
AGG	7.8	0	Icf	8.4	0	Drop3	8.1	0
Cnn	9.7	0	Rmhc	8.6	0	Icf	10.1	0
Drop3	9.8	0	Vsm	9.4	0	Ib2	10.5	0
Icf	11	0	Cnn	10.1	0	Vsm	10.6	0
Vsm	11.1	0	Ib2	10.5	0	Cnn	11.6	0

2.5.2.2. Resultados en Clasificación para Conjuntos de Tamaño Mediano

En la Sección 2.B ofrecemos los resultados para cada uno de los conjuntos de tamaño mediano evaluados. La Tabla 2.6 presenta los resultados medios y la Tabla 2.7 muestra la ordenación de los algoritmos por objetivo (reducción, acierto, o reducción-acierto):

Tabla 2.6: Resultados Medios de Selección de Prototipos en Clasificación en Conjuntos de Tamaño Mediano

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	46		94.19	94.18
Cnn (*)	4	97.32	79.33	80.17
Drop1 (*)	254	96.72	78.00	76.85
Drop2 (*)	215	89.57	88.62	88.62
Drop3 (*)	338	96.25	89.03	85.44
Enn	139	5.82	95.43	94.39
Ib2 (*)	2	97.78	75.24	75.84
Icf (*)	386	90.04	76.77	76.68
Mcs	101	2.66	95.96	94.38
Multied	1778	13.99	92.43	92.10
Renn	489	6.47	95.37	94.30
Rnn (*)	13017	96.88	81.27	81.74
Shrink	206	4.89	95.19	94.38
Vsm	94	1.04	94.16	94.17
Ib3 (*)	42	71.67	91.49	92.61
Rmhc (*)	34525	90.02	91.59	91.15
Ennsr (*)	37802	90.02	92.79	92.75
AGG	66157	62.53	94.74	93.85
AGE	54656	62.91	95.00	93.67
CHC (*)	8072	99.29	93.31	93.53
PBIL (*)	32942	73.13	96.23	94.13

Tabla 2.7: Ordenación de los Algoritmos por Objetivo en Clasificación para Conjuntos de Tamaño Mediano

% Reducción			% Test 1-NN			%Test 1-NN - %Reducción		
Nom	Ord	Mej	Nom	Ord	Mej	Nom	Ord	Mej
CHC	1	3	PBIL	2	2	CHC	1	3
Ib2	2	0	Ib3	3	1	Ennrs	4.3	0
Drop1	3.6	0	CHC	3.3	0	Drop3	4.6	0
Cnn	4	0	Ennrs	4.3	0	Rmhc	5.6	0
Drop3	5.3	0	Rmhc	6.3	0	Drop2	6.3	0
Rnn	5.3	0	Drop2	6.6	0	Ib3	6.6	0
Rmhc	8.3	0	Cnn	8	0	Drop1	7	0
Icf	8.6	0	Drop3	8.3	0	Cnn	7	0
Drop2	9	0	Ib2	8.3	0	Ib2	7.6	0
Ennrs	9.3	0	Rnn	8.6	0	Rnn	7.6	0
Ib3	9.6	0	Drop1	9.3	0	PBIL	9.6	0
PBIL	11.6	0	Icf	9.6	0	Icf	10.3	0

2.5.2.3. Análisis de los Resultados en Clasificación

Estudiando los resultados presentes en las Tablas 2.4 y 2.5, dedicadas a la evaluación de conjuntos de datos pequeños, podemos destacar:

- Los AAEE presentan los mejores porcentajes de reducción, conservando las tasas de acierto más elevadas, como podemos ver en la Tabla 2.4.
- En la Tabla 2.5 podemos ver que los AAEE son los que ofrecen los mejores resultados en reducción, ofreciendo el mejor comportamiento en 8 de los 10 conjuntos de datos. Del mismo modo, obtienen el mejor porcentaje de acierto en 8 de los conjuntos. Considerando el objetivo combinado, los AAEE continúan manteniendo el mejor comportamiento.

De este modo se puede concluir que los AAEE en SPP para clasificación aplicados sobre conjuntos de datos de tamaño pequeño ofrecen las mejores prestaciones. Son capaces de generar los mejores resultados tanto en cada uno de los objetivos independientes como en el combinado, manteniendo la homogeneidad de su comportamiento en cada uno de los conjuntos evaluados.

Considerando los resultados presentes en las Tablas 2.6 y 2.7 dedicadas a conjuntos de datos de tamaño mediano podemos observar lo siguiente:

- Los AAEE, y particularmente el CHC, aparecen como la mejor propuesta cuando crece el tamaño del problema. El algoritmo CHC ofrece los mejores porcentajes, tanto de reducción como de acierto.
- La Tabla 2.7 muestra que el algoritmo CHC es el que presenta el mejor comportamiento. Ofrece la mejor ordenación en la reducción y al considerar la combinación de ambos objetivos.
- En conjuntos de datos de este tamaño, tanto los AAEE como los algoritmos no evolutivos presentan dificultades para conseguir elevadas tasas de reducción manteniendo al mismo nivel las de acierto. Los algoritmos no evolutivos con porcentajes de reducción altos ven sensiblemente mermada su precisión.

Por tanto, los AAEE, concretamente el CHC, ofrecen las mejores prestaciones al ser aplicados sobre conjuntos de datos de tamaño mediano. Conservan su capacidad de reducción y de acierto de entre los conjuntos evaluados, independientemente de su tamaño, presentando elevados niveles en los mismos.

Los AAEE consiguen tanto en conjuntos de datos pequeños como medianos equilibrar de manera adecuada la reducción con la precisión, llevando a cabo la selección más eficaz.

2.5.3. Resultados en Selección de Conjuntos de Entrenamiento

En esta sección incluimos los resultados en la SPP aplicada a la selección de conjuntos de entrenamiento, estudiando el comportamiento de las diferentes técnicas al aplicarlas sobre conjuntos de diferente tamaño.

2.5.3.1. Resultados en Selección de Conjuntos de Entrenamiento para Conjuntos de Tamaño Pequeño

En la Sección 2.C ofrecemos los resultados para cada uno de los conjuntos de tamaño pequeño evaluados. La Tabla 2.8 muestra los resultados medios y la Tabla 2.9 ofrece la ordenación de los algoritmos por objetivo (reducción, acierto, o reducción-acierto):

Tabla 2.8: Resultados Medios Seleccionando Conjuntos de Entrenamiento en Conjuntos de Tamaño Pequeño

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0.01		89.53	71.79
Cnn (*)	0.01	71.98	85.45	49.10
Drop1(*)	0.23	86.97	87.25	53.41
Drop2	0.2	55.19	83.60	62.33
Drop3(*)	0.15	78.56	86.79	55.52
Enn	0.1	35.07	95.66	68.29
Ib2 (*)	0.03	77.80	84.71	51.22
Icf (*)	0.29	75.81	94.59	64.02
Mcs	0.09	16.90	90.70	71.82
Multied	0.24	54.70	99.83	60.25
Renn	0.25	37.43	97.27	68.40
Rnn (*)	1.89	90.38	92.34	62.03
Shrink	0.08	30.41	96.12	69.26
Vsm (*)	0.01	74.56	90.31	63.25
Ib3	0.06	65.71	79.18	65.22
Rmhc(*)	54.37	90.27	89.13	64.70
Ennrs(*)	69.95	90.27	91.31	62.04
AGG (*)	70.8	87.72	90.29	62.87
AGE (*)	68.6	90.50	88.94	63.36
CHC (*)	20.48	96.05	80.49	53.61
PBIL (*)	43.2	93.79	93.25	67.44

Tabla 2.9: Ordenación de los Algoritmos por Objetivo en Selección de Conjuntos de Entrenamiento para Conjuntos de Tamaño Pequeño

%Reducción			% Test C4.5			%Test C4.5 - %Reducción		
Nom	Ord	Mej	Nom	Ord	Mej	Nom	Ord	Mej
CHC	1.3	7	AGE	4.7	1	PBIL	2.4	3
PBIL	2.9	1	PBIL	4.9	2	AGE	4.3	1
AGE	5.1	0	Icf	5	1	Ennrs	4.7	1
Rnn	5.8	0	Ennrs	5.3	0	Rnn	4.8	0
Rmhc	5.8	1	Rmhc	5.5	2	Rmhc	4.8	3
Drop1	6.6	1	AGG	5.5	2	AGG	5.9	1
Ennrs	6.8	0	Rnn	6.1	1	CHC	6.1	1
Ib2	7.3	0	Vsm	7.2	0	Drop1	8.5	0
AGG	7.8	0	Drop1	8.3	1	Icf	8.5	0
Cnn	9.7	0	Drop3	8.7	0	Vsm	9.3	0
Drop3	9.8	0	Ib2	9.9	0	Drop3	9.9	0
Icf	11	0	CHC	9.9	0	Ib2	10.5	0
Vsm	11.1	0	Cnn	10	0	Cnn	11.3	0

2.5.3.2. Resultados en Selección de Conjuntos de Entrenamiento para Conjuntos de Tamaño Mediano

En la Sección 2.D ofrecemos los resultados para cada uno de los conjuntos de tamaño pequeño evaluados. La Tabla 2.10 muestra los resultados medios y la Tabla 2.11 presenta la ordenación de los algoritmos por objetivo (reducción, acierto, o reducción-acierto):

Tabla 2.10: Resultados Medios Seleccionando Conjuntos de Entrenamiento en Conjuntos de Tamaño Pequeño

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	1		98.82	94.07
Cnn (*)	4	97.32	93.68	72.03
Drop1 (*)	254	96.72	95.19	78.40
Drop2 (*)	215	89.57	95.42	79.04
Drop3 (*)	338	96.25	95.14	75.33
Enn	139	5.82	99.26	93.83
Ib2 (*)	2	97.78	93.06	68.13
Icf (*)	386	90.04	97.85	82.39
Mcs	101	2.66	98.97	93.91
Multied	1778	13.99	99.77	90.61
Renn	489	6.47	99.31	93.83
Rnn (*)	13017	96.88	99.16	69.50
Shrink	206	4.89	99.23	93.85
Vsm	94	1.04	98.84	94.07
Ib3 (*)	42	71.67	94.94	84.67
Rmhc (*)	34525	90.02	98.11	91.80
Ennsr (*)	37802	90.02	97.99	89.73
AGG	66157	62.53	98.72	92.60
AGE	54656	62.91	98.38	92.55
CHC (*)	8072	99.29	98.29	91.82
PBIL (*)	32942	73.13	98.71	92.72

Tabla 2.11: Ordenación de los Algoritmos por Objetivo en Selección de Conjuntos de Entrenamiento para Conjuntos de Tamaño Mediano

%Reducción			% Test C4.5			%Test C4.5 - %Reducción		
Nom	Ord	Mej	Nom	Ord	Mej	Nom	Ord	Mej
CHC	1	3	PBIL	1.3	2	CHC	1	3
Ib2	2	0	CHC	2.6	0	Rmhc	4.3	0
Drop1	3.6	0	Rmhc	3	1	Ennrs	4.6	0
Cnn	4	0	Ennrs	4	0	Drop1	5.6	0
Drop3	5.3	0	Ib3	4	0	Drop3	6	0
Rnn	5.3	0	Icf	6.3	0	Icd	6.3	0
Rmhc	8.3	0	Drop2	7	0	Ib3	7.3	0
Icf	8.6	0	Drop1	8.6	0	Cnn	7.6	0
Drop2	9	0	Drop3	9.3	0	PBIL	8	0
Ennrs	9.3	0	Cnn	9.6	0	Drop2	8.6	0
Ib3	9.6	0	Rnn	10.6	0	Rnn	9	0
PBIL	11.6	0	Ib2	11.3	0	Ib2	9.3	0

2.5.3.3. Análisis de Resultados en Selección de Conjuntos de Entrenamiento

Estudiando los resultados presentes en las Tablas 2.8 y 2.9, dedicadas a la evaluación de conjuntos de datos pequeños, podemos destacar:

- La Tabla 2.8 muestra que los `AAEE` son tan solo mejorados en precisión por aquellas técnicas que ofrecen tasas de reducción muy inferiores.
- En la Tabla 2.9 aparecen los `AAEE` como los mejores con respecto a la precisión, proporcionando el mejor resultados en 5 de los 10 conjuntos. Los `AAEE` son de nuevo aquellos que ofrecen el mejor equilibrio entre ambos objetivos, proporcionando el mejor comportamiento en 6 de las ocasiones.

De este modo se puede concluir que los `AAEE` son capaces de seleccionar el conjunto de entrenamiento más prometedor cuando son aplicados en conjuntos de datos de tamaño pequeño.

En el caso de ser evaluados sobre conjuntos de datos de tamaño mediano, cuyos resultados podemos ver en las Tablas 2.10 y 2.11, podemos destacar:

- El algoritmo `CHC` es el que ofrece el mejor comportamiento en selección de conjuntos de entrenamiento. Ofrece los mejores porcentajes de reducción

junto con las mayores tasas de precisión. La tercera columna de la tabla 2.11 muestra su capacidad de reducción. El algoritmo CHC consigue los mejores resultados en el objetivo combinado en todos los conjuntos de datos estudiados. Dicha circunstancia destaca al algoritmo CHC como el mejor de la comparativa.

- En este tipo de conjuntos de datos, los algoritmos no evolutivos seleccionan conjuntos de entrenamiento que no son capaces de generalizar adecuadamente. Esta situación aparece claramente reflejada en el caso de Ib2 y Rnn en la Tabla 2.10, en la que la diferencia entre los porcentajes de acierto en entrenamiento y test es significativa.
- Los dos algoritmos genéticos clásicos no presentan el mejor comportamiento en conjuntos de tamaño mediano. Esta circunstancia será estudiada en la siguiente sección.

El algoritmo CHC aparece claramente como aquel que realiza la selección de conjuntos de entrenamiento en conjuntos de tamaño mediano y pequeño de manera más eficaz, ofreciendo el mejor equilibrio entre reducción y acierto de manera homogénea.

2.6. Análisis de los Algoritmos Evolutivos en Selección de Prototipos

En esta sección estudiaremos algunos aspectos del comportamiento de los AAEE en SPP, analizando su mecanismo de selección junto con sus tiempos de ejecución asociados. Para ello, dedicaremos la Subsección 2.6.1 a estudiar los tiempos de ejecución que presentan. La Subsección 2.6.2 analizará el mecanismo de selección empleado en los AAEE, comparándolo con el de otras técnicas no evolutivas.

2.6.1. Tiempos de Ejecución

El tiempo de ejecución de los AAEE es superior al de los algoritmos no evolutivos, debido al proceso de evolución que tienen asociado.

Sin embargo, su empleo es más que interesante debido a que los algoritmos no evolutivos, que presentan un menor coste computacional, son mucho menos eficaces en la selección. En caso de presentar una precisión elevada, su reducción es prácticamente nula, y viceversa.

Dado que la selección de las muestras mas representativas se lleva a cabo una única vez para obtener el conjunto seleccionado (no es necesario un proceso contínuo de selección) es mucho más interesante la eficacia del proceso que la eficiencia.

Por tanto, el algoritmo que presenta el mejor comportamiento en este sentido es el CHC, presentando el menor coste computacional de entre todos los AAEE y probabilísticos, e incluso que algún algoritmo no evolutivo, y la mayor eficacia de entre todos ellos. Su menor coste computacional de entre los AAEE es debido a las capacidades a nivel de reducción que ofrece. La diversidad que introduce permite la mayor exploración del dominio de búsqueda, consiguiendo más rápidamente soluciones compuestas por un menor número de muestras.

En los experimentos se ha comprobado que el algoritmo CHC es capaz de generar cromosomas que seleccionan un reducido número de instancias desde el comienzo del proceso de evolución, lo que reduce el tiempo asociado al calculo de la precisión para la función objetivo.

2.6.2. Análisis del Mecanismo de Selección de los Algoritmos Evolutivos

Para poder evaluar la estrategia de selección de los algoritmos estudiados, proyectaremos sus subconjuntos de datos seleccionados para el conjunto de datos Iris. Para ello consideraremos sus dos caracteres más discriminantes (longitud y anchura del pétalo [LS95]).

La elección de este conjunto se debe a que es uno de los más conocidos así como su posibilidad de proyección empleando representación gráfica en dos dimensiones, para poder estudiar la distribución de las instancias seleccionadas.

Hemos comparado el mecanismo de selección de los siguientes algoritmos:

- `Multiedit`: Escogido dado que ofrece una precisión elevada y pequeña tasa de reducción de entre las técnicas no evolutivas.
- `Cnn`, `Ib2` and `Drop2`: Seleccionados por ser los algoritmos no evolutivos que proporcionan las mejores tasas de reducción.
- `Ib3`: Es la técnica no evolutiva que presenta el mejor equilibrio entre reducción y precisión.
- `CHC`: Este es el algoritmo evolutivo que mejor comportamiento ofrece de entre todos los estudiados.

Incluiremos así mismo la proyección del conjunto Iris completo para poder observar la distribución de las muestras sin llevar a cabo selección alguna.

El conjunto de Figuras de la 2.5 a la 2.11 muestran gráficamente la estrategia de selección seguida por cada algoritmo.

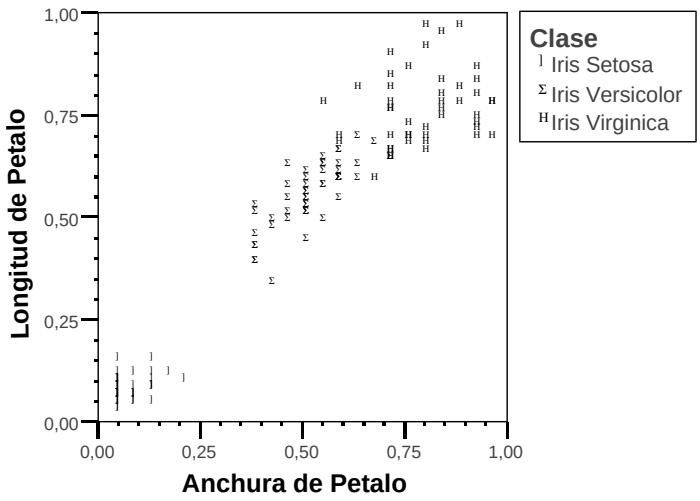

Figura 2.5: Iris completo

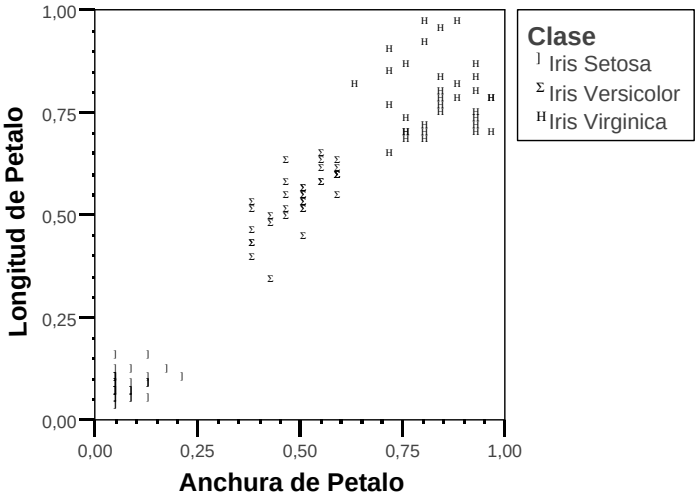

Figura 2.6: Selección mediante Multiedit en Iris

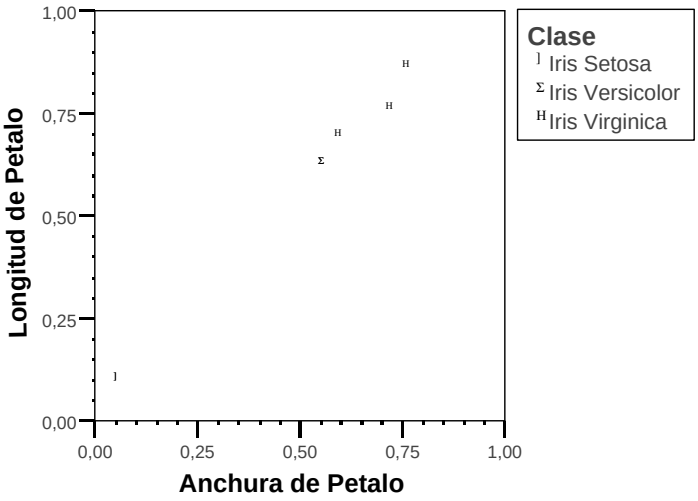

Figura 2.7: Selección mediante Cnn en Iris

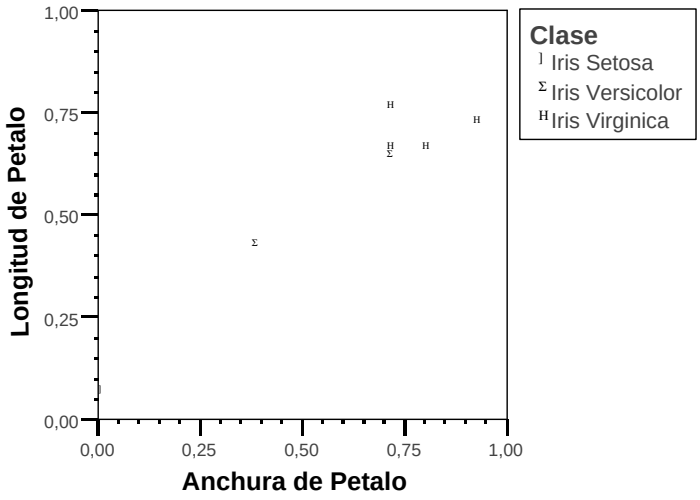

Figura 2.8: Selección mediante Ib2 en Iris

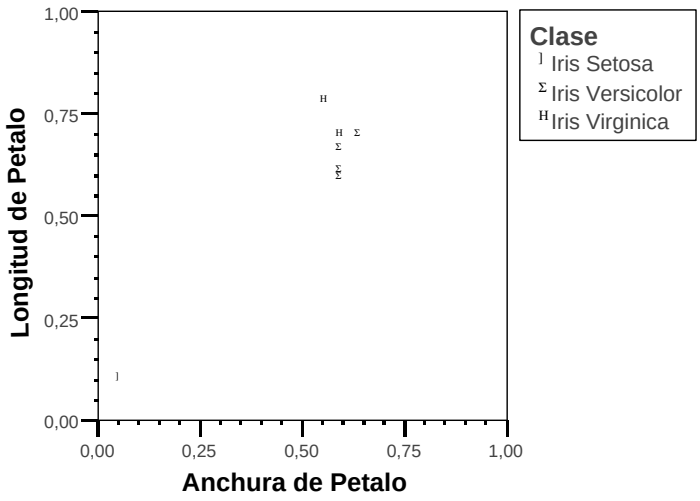

Figura 2.9: Selección mediante Drop2 en Iris

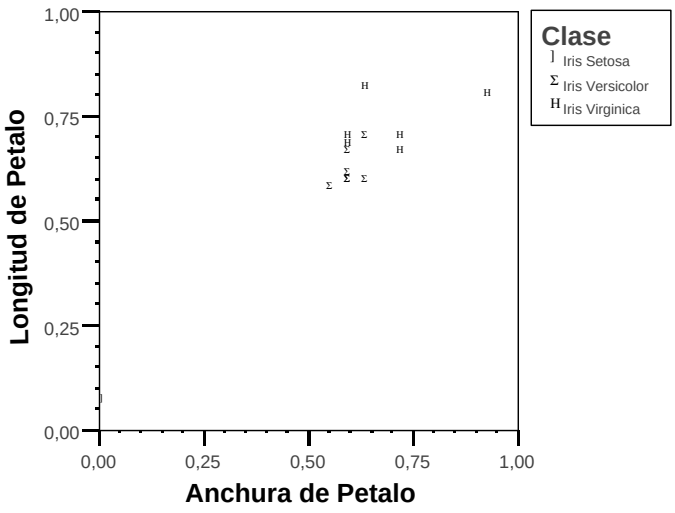

Figura 2.10: Selección mediante Ib3 en Iris

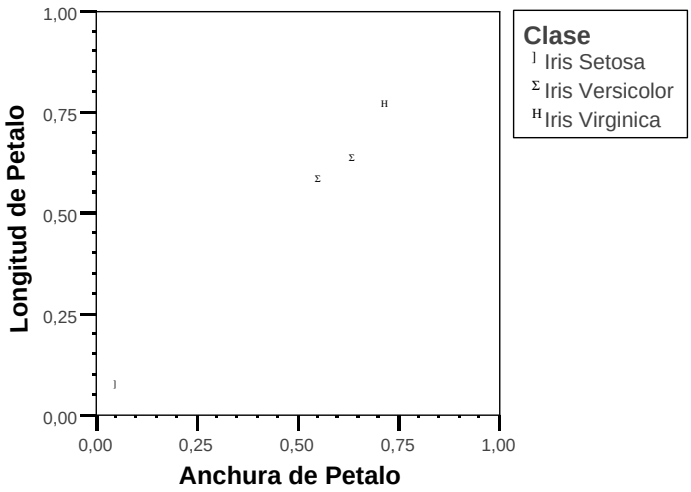

Figura 2.11: Selección mediante CHC en Iris

Analizando las anteriores figuras podemos destacar los siguientes comportamientos:

- `Multiedit`: Este algoritmo conserva aquellas instancias situadas en el centro de las clases, eliminando aquellas presentes en las fronteras.
- `Cnn`, `Ib2` and `Drop2`: Estas técnicas tienden a mantener las instancias situadas en las fronteras. Su heurística se basa en eliminar las instancias internas a las clases dado que pueden ser clasificadas empleando aquellas situadas en las fronteras. El problema que supone esta heurística es el clasificar y seleccionar correctamente las instancias situadas en las fronteras entre clases.
- `Ib3`: Presenta una heurística similar a la anterior, sin embargo selecciona una mayor cantidad de instancias fronterizas. Debido al equilibrio reducción-acierto que ofrece, pretende conservar información suficiente de una zona del espacio de tal forma que la reducción no afecte a la precisión. Esto supone que en algunos caso se conserven más instancias que las estrictamente necesarias.
- `CHC`: En este caso se combinan muestras tanto internas como fronterizas en la selección. No se tiende a escoger las instancias dependiendo de su posición en el espacio de búsqueda. El algoritmo `CHC` selecciona aquellas instancias que incrementen la capacidad de precisión del conjunto al completo, lo cual significa que se seleccionan aquellas muestras más representativas independientemente de su posición.

Esta es la razón por la que el algoritmo `CHC` ofrece el mejor equilibrio entre reducción y acierto. `CHC` no tiene limitada su capacidad de selección por una heurística o estrategia particular, de tal forma que puede dedicar sus esfuerzos en reducir sin afectar sensiblemente a la precisión.

2.7. Comentarios Finales

En este capítulo abordamos el empleo de AAEE en SII y su uso para RDD en el proceso de descubrimiento de información.

Se ha efectuado un estudio experimental en el que se han comparado cuatro AAEE frente a algoritmos de selección no evolutivos, siguiendo una doble perspectiva: clasificación y selección de conjuntos de entrenamiento. El estudio se ha llevado a cabo empleando conjuntos de datos de diferente tamaño (pequeño y mediano).

Las principales conclusiones alcanzadas son las siguientes:

- Los AAEE mejoran el comportamiento de los algoritmos no evolutivos, ofreciendo simultáneamente dos ventajas: mayor reducción del conjunto de datos y precisión elevada.
- El algoritmo CHC es la técnica más destacada de acuerdo a la comparativa desarrollada. Ofrece el mejor comportamiento tanto en clasificación como en selección de conjuntos de entrenamiento, independientemente del tamaño del conjunto de datos considerado. Como añadido, se destaca por presentar unas prestaciones tales que hacen que sea el algoritmo evolutivo más eficiente en los problemas considerados en nuestro estudio.
- En conjuntos de datos de tamaño mediano, los algoritmos no evolutivos no presentan un comportamiento equilibrado entre reducción y acierto. En caso de presentar una elevada reducción, la precisión es pobre. En caso de destacar su precisión, no se lleva a cabo ningún tipo de reducción.
- El mecanismo de selección ciega (no dirigida por heurísticas) seguido por el algoritmo CHC permite escoger a las muestras más representativas del conjunto independientemente de su posición en el espacio de búsqueda.

De esta forma, como conclusión final y tras el estudio y análisis efectuado, se consideran los AAEE como el mecanismo de SII más adecuado, destacándose notablemente el algoritmo CHC frente al resto. Ha demostrado ser una herramienta potente para obtener los menores subconjuntos de prototipos con las muestras más representativas. El algoritmo CHC lleva a cabo dicho proceso de selección buscando maximizar los objetivos de precisión y reducción perseguidos, independientemente de heurísticas asociadas a distribución de instancias.

2.A Tablas de Resultados de Conjunto de Datos de Tamaño Pequeño en Clasificación

Tabla 2.12: Selección de Prototipos aplicada en Cleveland para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		52.42 %	53.90 %
Cnn	0	64.57 %	27.79 %	32.62 %
Drop1	0	82.45 %	62.25 %	54.37 %
Drop2	0	38.61 %	51.65 %	51.65 %
Drop3	0	77.80 %	67.17 %	49.14 %
Enn	0	47.58 %	62.48 %	54.69 %
Ib2	0	83.87 %	17.83 %	20.68 %
Icf	0	79.84 %	53.07 %	46.11 %
Mcs	0	28.88 %	63.57 %	53.31 %
Multied	0	61.61 %	53.87 %	54.07 %
Renn	0	50.43 %	62.52 %	55.78 %
Rnn	3	88.29 %	56.16 %	56.12 %
Shrink	0	43.54 %	63.60 %	54.34 %
Vsm	0	79.74 %	54.10 %	53.71 %
Ib3	0	47.48 %	43.42 %	51.52 %
Rmhc	34	90.27 %	37.56 %	38.47 %
Ennrs	42	90.27 %	54.31 %	43.40 %
AGG	45	89.27 %	66.11 %	55.29 %
AGE	41	90.87 %	65.74 %	55.06 %
CHC	11	98.02 %	57.73 %	51.56 %
PBIL	31	96.45 %	57.73 %	51.58 %

Tabla 2.13: Selección de Prototipos aplicada en Glass para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		73.11 %	75.58 %
Cnn	0	70.35 %	46.28 %	52.34 %
Drop1	0	81.36 %	73.62 %	71.47 %
Drop2	0	55.04 %	71.98 %	71.98 %
Drop3	0	75.54 %	86.14 %	67.05 %
Enn	0	27.62 %	76.53 %	73.41 %
Ib2	0	80.68 %	39.36 %	47.24 %
Icf	0	67.28 %	65.65 %	65.34 %
Mcs	0	13.50 %	79.13 %	73.40 %
Multied	0	67.54 %	52.60 %	50.89 %
Renn	0	31.77 %	75.86 %	71.66 %
Rnn	2	82.14 %	73.31 %	67.09 %
Shrink	0	26.59 %	74.98 %	71.11 %
Vsm	0	76.42 %	54.36 %	55.02 %
Ib3	0	61.74 %	59.45 %	69.46 %
Rmhc	25	90.27 %	55.60 %	59.96 %
Ennrs	34	90.27 %	68.92 %	62.12 %
AGG	36	85.93 %	78.82 %	67.96 %
AGE	37	86.45 %	80.07 %	68.72 %
CHC	8	92.99 %	76.79 %	59.19 %
PBIL	22	91.22 %	77.88 %	67.62 %

Tabla 2.14: Selección de Prototipos aplicada en Iris para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		95.26 %	95.33 %
Cnn	0	92.59 %	87.63 %	90.00 %
Drop1	0	94.52 %	84.44 %	83.33 %
Drop2	0	88.59 %	92.00 %	92.00 %
Drop3	0	94.09 %	96.74 %	95.33 %
Enn	0	4.74 %	95.93 %	96.00 %
Ib2	0	94.81 %	86.44 %	89.33 %
Icf	0	64.07 %	89.26 %	88.00 %
Mcs	0	2.59 %	95.33 %	95.33 %
Multied	0	16.22 %	95.78 %	94.67 %
Renn	0	4.74 %	95.93 %	96.00 %
Rnn	0	91.70 %	95.93 %	94.67 %
Shrink	0	4.74 %	95.93 %	95.33 %
Vsm	0	76.30 %	94.89 %	94.00 %
Ib3	0	89.48 %	91.26 %	94.00 %
Rmhc	7	90.27 %	89.93 %	89.33 %
Ennrs	10	90.27 %	98.77 %	97.78 %
AGG	11	94.81 %	98.37 %	94.00 %
AGE	11	95.11 %	98.44 %	95.33 %
CHC	2	96.30 %	98.22 %	93.33 %
PBIL	8	96.52 %	97.70 %	98.00 %

Tabla 2.15: Selección de Prototipos aplicada en Led24Digit para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		40.49 %	42.61 %
Cnn	0	30.61 %	27.67 %	36.58 %
Drop1	0	76.38 %	52.50 %	33.53 %
Drop2	0	22.77 %	39.08 %	39.08 %
Drop3	0	49.75 %	85.24 %	29.89 %
Enn	0	59.51 %	50.78 %	39.12 %
Ib2	0	43.61 %	26.62 %	36.62 %
Icf	0	67.18 %	49.67 %	38.62 %
Mcs	0	31.61 %	54.10 %	41.07 %
Multied	0	93.34 %	15.82 %	16.11 %
Renn	0	66.00 %	48.67 %	36.87 %
Rnn	1	86.22 %	42.34 %	33.24 %
Shrink	0	53.61 %	50.28 %	40.65 %
Vsm	0	62.31 %	32.45 %	31.02 %
Ib3	0	33.99 %	31.00 %	37.15 %
Rmhc	9	90.27 %	27.56 %	28.89 %
Ennrs	14	90.27 %	40.55 %	33.41 %
AGG	13	83.44 %	58.32 %	34.69 %
AGE	13	83.67 %	59.61 %	37.96 %
CHC	10	91.55 %	55.00 %	25.66 %
PBIL	3	87.11 %	57.39 %	39.51 %

Tabla 2.16: Selección de Prototipos aplicada en Led7Digit para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		54.87 %	20.02 %
Cnn	0	36.83 %	37.41 %	36.98 %
Drop1	0	98.69 %	23.73 %	23.49 %
Drop2	0	32.94 %	50.82 %	50.82 %
Drop3	0	88.25 %	72.05 %	26.89 %
Enn	0	79.42 %	28.41 %	27.61 %
Ib2	0	27.50 %	23.97 %	24.43 %
Icf	0	81.46 %	24.07 %	23.58 %
Mcs	0	6.09 %	34.71 %	34.90 %
Multied	0	84.09 %	24.49 %	25.00 %
Renn	0	80.17 %	28.41 %	27.41 %
Rnn	3	96.05 %	34.00 %	34.02 %
Shrink	1	68.48 %	28.83 %	28.19 %
Vsm	0	71.76 %	22.83 %	23.05 %
Ib3	0	64.72 %	38.46 %	41.31 %
Rmhc	51	90.27 %	35.89 %	37.40 %
Ennrs	79	90.27 %	59.01 %	59.00 %
AGG	73	89.58 %	44.34 %	45.13 %
AGE	75	93.69 %	45.23 %	42.63 %
CHC	23	96.93 %	76.31 %	61.13 %
PBIL	45	95.64 %	36.31 %	33.40 %

Tabla 2.17: Selección de Prototipos aplicada en Lymphography para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		34.84 %	36.81 %
Cnn	0	55.99 %	22.45 %	27.71 %
Drop1	0	80.65 %	48.65 %	40.74 %
Drop2	0	20.20 %	39.23 %	39.23 %
Drop3	0	58.35 %	84.92 %	40.10 %
Enn	0	65.16 %	49.47 %	37.69 %
Ib2	0	64.06 %	23.79 %	31.71 %
Icf	0	73.50 %	47.97 %	39.14 %
Mcs	0	41.15 %	51.80 %	36.23 %
Multied	0	88.94 %	45.27 %	45.46 %
Renn	0	70.35 %	49.78 %	41.10 %
Rnn	0	89.79 %	45.05 %	46.08 %
Shrink	0	57.96 %	50.30 %	40.96 %
Vsm	0	68.50 %	39.20 %	38.30 %
Ib3	0	36.72 %	26.71 %	33.88 %
Rmhc	8	90.27 %	37.32 %	37.25 %
Ennrs	12	90.27 %	53.77 %	44.05 %
AGG	11	88.36 %	57.06 %	46.34 %
AGE	12	89.50 %	58.27 %	41.49 %
CHC	7	92.50 %	62.76 %	46.07 %
PBIL	9	91.52 %	62.31 %	47.96 %

Tabla 2.18: Selección de Prototipos aplicada en Monk para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		70.71 %	72.46 %
Cnn	0	90.33 %	50.23 %	51.16 %
Drop1	0	77.91 %	82.43 %	71.52 %
Drop2	0	58.54 %	71.07 %	71.07 %
Drop3	0	70.76 %	93.78 %	71.30 %
Enn	0	29.29 %	89.63 %	69.48 %
Ib2	0	98.23 %	34.29 %	35.86 %
Icf	0	69.26 %	71.73 %	63.20 %
Mcs	0	25.39 %	89.92 %	72.46 %
Multied	0	52.29 %	67.13 %	67.14 %
Renn	0	30.25 %	88.91 %	64.84 %
Rnn	3	88.97 %	71.68 %	67.14 %
Shrink	0	18.70 %	84.49 %	69.70 %
Vsm	0	81.53 %	69.86 %	69.24 %
Ib3	0	67.31 %	68.96 %	70.39 %
Rmhc	79	90.23 %	60.67 %	62.05 %
Ennrs	91	90.23 %	68.47 %	63.57 %
AGG	88	81.48 %	88.09 %	71.75 %
AGE	90	82.90 %	88.79 %	72.44 %
CHC	19	97.74 %	74.97 %	72.21 %
PBIL	28	93.11 %	78.52 %	64.54 %

Tabla 2.19: Selección de Prototipos aplicada en Pima para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		70.53 %	68.49 %
Cnn	0	94.40 %	44.82 %	45.18 %
Drop1	0	85.00 %	76.88 %	70.32 %
Drop2	0	52.07 %	69.28 %	69.28 %
Drop3	0	82.46 %	89.35 %	69.14 %
Enn	0	29.47 %	79.50 %	73.31 %
Ib2	0	93.87 %	43.10 %	44.03 %
Icf	0	77.17 %	70.99 %	67.07 %
Mcs	0	16.45 %	82.38 %	71.23 %
Multied	0	57.74 %	66.85 %	66.16 %
Renn	0	32.57 %	78.99 %	73.70 %
Rnn	5	90.73 %	70.69 %	66.94 %
Shrink	2	24.07 %	78.31 %	73.97 %
Vsm	0	77.56 %	68.89 %	66.41 %
Ib3	0	70.17 %	61.70 %	62.90 %
Rmhc	89	90.02 %	65.78 %	68.37 %
Ennrs	101	90.02 %	74.49 %	70.88 %
AGG	108	80.92 %	82.78 %	68.89 %
AGE	105	89.64 %	84.72 %	70.06 %
CHC	35	98.29 %	78.78 %	73.57 %
PBIL	58	93.06 %	79.90 %	74.49 %

Tabla 2.20: Selección de Prototipos aplicada en Wine para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		96.07 %	96.08 %
Cnn	0	89.51 %	89.76 %	92.12 %
Drop1	0	93.82 %	85.33 %	87.12 %
Drop2	0	90.76 %	93.29 %	93.29 %
Drop3	0	90.78 %	93.18 %	94.44 %
Enn	0	3.93 %	96.07 %	95.55 %
Ib2	0	93.76 %	89.20 %	90.41 %
Icf	0	83.02 %	96.69 %	96.63 %
Mcs	0	1.00 %	96.63 %	96.08 %
Multied	0	17.79 %	94.69 %	94.44 %
Renn	0	3.93 %	96.07 %	95.55 %
Rnn	1	91.01 %	96.13 %	93.86 %
Shrink	0	3.93 %	96.07 %	95.55 %
Vsm	0	80.82 %	76.63 %	72.74 %
Ib3	0	89.26 %	89.70 %	91.66 %
Rmhc	19	90.01 %	64.30 %	65.20 %
Ennrs	23	90.00 %	79.17 %	68.52 %
AGG	25	93.63 %	99.19 %	95.00 %
AGE	27	95.00 %	99.31 %	97.19 %
CHC	9	96.82 %	80.15 %	65.78 %
PBIL	19	94.94 %	79.28 %	71.96 %

Tabla 2.21: Selección de Prototipos aplicada en Wisconsin para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	0		96.06 %	96.05 %
Cnn	0	94.65 %	57.02 %	57.11 %
Drop1	0	98.96 %	91.18 %	91.51 %
Drop2	0	92.37 %	93.86 %	93.86 %
Drop3	0	97.85 %	97.65 %	94.73 %
Enn	0	3.94 %	97.93 %	96.79 %
Ib2	0	97.58 %	92.86 %	93.27 %
Icf	0	95.27 %	91.67 %	92.39 %
Mcs	0	2.33 %	98.02 %	96.79 %
Multied	0	7.40 %	96.83 %	96.64 %
Renn	0	4.13 %	97.89 %	96.79 %
Rnn	4	98.93 %	96.36 %	95.90 %
Shrink	2	2.46 %	97.69 %	96.64 %
Vsm	0	70.70 %	85.65 %	85.21 %
Ib3	0	96.18 %	93.52 %	94.43 %
Rmhc	73	90.08 %	95.57 %	94.73 %
Ennrs	79	90.08 %	97.78 %	98.54 %
AGG	82	89.75 %	98.08 %	96.34 %
AGE	78	98.21 %	98.70 %	95.65 %
CHC	24	99.35 %	97.93 %	95.17 %
PBIL	61	98.36 %	97.89 %	97.21 %

2.B Tablas de Resultados de Conjunto de Datos de Tamaño Mediano en Clasificación

Tabla 2.22: Selección de Prototipos aplicada en Pen-Based Recognition para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	66		99.36 %	99.39 %
Cnn	4	98.04 %	84.85 %	85.69 %
Drop1	374	98.45 %	86.23 %	86.02 %
Drop2	318	97.69 %	91.03 %	91.03 %
Drop3	391	98.07 %	90.33 %	90.05 %
Enn	269	0.64 %	99.40 %	99.31 %
Ib2	2	98.49 %	74.20 %	75.04 %
Icf	537	92.42 %	89.79 %	89.51 %
Mcs	141	0.36 %	99.54 %	99.40 %
Multied	4274	6.47 %	97.78 %	97.64 %
Renn	579	0.66 %	99.40 %	99.31 %
Rnn	33168	97.52 %	77.97 %	79.03 %
Shrink	277	0.64 %	99.39 %	99.31 %
Vsm	107	0.00 %	99.36 %	99.39 %
Ib3	9	96.42 %	96.73 %	98.00 %
Rmhc	69802	90.02 %	97.43 %	97.19 %
Ennrs	75988	90.02 %	98.50 %	98.01 %
AGG	149281	61.47 %	99.22 %	98.82 %
AGE	103102	61.77 %	99.35 %	99.00 %
CHC	18845	98.99 %	96.29 %	98.94 %
PBIL	83923	73.59 %	99.77 %	99.05 %

Tabla 2.23: Selección de Prototipos aplicada en SatImage para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	36.20		90.33 %	90.41 %
Cnn	5	95.93 %	60.63 %	61.96 %
Drop1	206	93.66 %	84.29 %	81.68 %
Drop2	183	83.49 %	83.45 %	83.45 %
Drop3	301	93.25 %	87.93 %	81.03 %
Enn	79	9.67 %	92.11 %	90.10 %
Ib2	3	96.75 %	59.00 %	59.59 %
Icf	378	84.53 %	70.18 %	70.18 %
Mcs	92	4.31 %	93.29 %	90.32 %
Multied	836	24.82 %	86.83 %	86.08 %
Renn	490	11.05 %	92.10 %	89.87 %
Rnn	4042	95.15 %	73.26 %	73.68 %
Shrink	184	8.86 %	91.59 %	90.07 %
Vsm	99	0.00 %	90.33 %	90.41 %
Ib3	22	84.66 %	84.51 %	86.45 %
Rmhc	15879	90.02 %	86.13 %	85.29 %
Ennrs	16075	90.02 %	88.08 %	88.29 %
AGG	20245	63.07 %	91.22 %	89.94 %
AGE	21512	63.35 %	91.76 %	89.32 %
CHC	2479	99.06 %	89.45 %	89.67 %
PBIL	6917	72.19 %	93.75 %	90.48 %

Tabla 2.24: Selección de Prototipos aplicada en Thyroid para Clasificación

	Tpo	Red	%Ac Trn	%Ac Test
1-NN	36.2		92.87 %	92.74 %
Cnn	3	98.00 %	92.50 %	92.86 %
Drop1	182	98.06 %	63.47 %	62.86 %
Drop2	143	87.54 %	91.37 %	91.37 %
Drop3	322	97.44 %	88.82 %	85.24 %
Enn	68	7.15 %	94.78 %	93.76 %
Ib2	2	98.11 %	92.53 %	92.89 %
Icf	244	93.17 %	70.34 %	70.35 %
Mcs	71	3.30 %	95.05 %	93.42 %
Multied	224	10.69 %	92.67 %	92.59 %
Renn	399	7.69 %	94.62 %	93.71 %
Rnn	1841	97.98 %	92.58 %	92.51 %
Shrink	156	5.18 %	94.58 %	93.76 %
Vsm	76	3.12 %	92.80 %	92.71 %
Ib3	94	33.93 %	93.22 %	93.38 %
Rmhc	17895	90.03 %	91.22 %	90.98 %
Ennrs	21342	90.03 %	91.78 %	91.95 %
AGG	28945	63.05 %	93.78 %	92.79 %
AGE	39354	63.60 %	93.90 %	92.69 %
CHC	2891	99.83 %	94.20 %	91.98 %
PBIL	7985	73.61 %	95.17 %	92.86 %

2.C Tablas de Resultados de Conjunto de Datos de Tamaño Pequeño en Selección de Conjuntos de Entrenamiento

Tabla 2.25: Selección de Conjuntos de Entrenamiento aplicada en Cleveland

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		84.20 %	65.60 %
Cnn	0	64.57 %	80.90 %	30.10 %
Drop1	0	82.45 %	90.29 %	53.18 %
Drop2	0	38.61 %	81.83 %	48.38 %
Drop3	0	77.80 %	93.64 %	51.97 %
Enn	0	47.58 %	95.93 %	53.43 %
Ib2	0	83.87 %	78.74 %	44.73 %
Icf	0	79.84 %	91.48 %	53.18 %
Mcs	0	28.88 %	88.86 %	54.11 %
Multied	0	61.61 %	100.00 %	54.04 %
Renn	0	50.43 %	96.98 %	56.11 %
Rnn	3	88.29 %	96.29 %	55.41 %
Shrink	0	43.54 %	93.84 %	50.66 %
Vsm	0	79.74 %	96.03 %	51.56 %
Ib3	0	47.48 %	80.52 %	50.43 %
Rmhc	34	90.27 %	84.87 %	30.40 %
Ennrs	42	90.27 %	90.41 %	54.36 %
AGG	41	89.27 %	87.32 %	57.65 %
AGE	45	90.87 %	83.75 %	54.05 %
CHC	11	98.02 %	80.33 %	48.66 %
PBIL	31	96.45 %	92.37 %	47.65 %

Tabla 2.26: Selección de Conjuntos de Entrenamiento aplicada en Glass

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		92.70 %	60.90 %
Cnn	0	70.35 %	86.26 %	44.39 %
Drop1	0	81.36 %	95.79 %	52.97 %
Drop2	0	55.04 %	89.28 %	59.21 %
Drop3	0	75.54 %	91.93 %	54.26 %
Enn	0	27.62 %	96.35 %	66.37 %
Ib2	0	80.68 %	89.00 %	45.00 %
Icf	0	67.28 %	93.01 %	59.81 %
Mcs	0	13.50 %	93.04 %	66.67 %
Multied	0	67.54 %	99.70 %	50.50 %
Renn	0	31.77 %	96.65 %	65.01 %
Rnn	2	82.14 %	91.02 %	55.61 %
Shrink	0	26.59 %	96.11 %	69.04 %
Vsm	0	76.42 %	90.36 %	48.09 %
Ib3	0	61.74 %	84.52 %	57.76 %
Rmhc	25	90.27 %	91.61 %	68.00 %
Ennrs	34	90.27 %	87.93 %	57.96 %
AGG	36	85.93 %	96.15 %	58.00 %
AGE	37	86.45 %	90.63 %	57.55 %
CHC	8	92.99 %	81.64 %	40.14 %
PBIL	22	91.22 %	90.21 %	57.75 %

Tabla 2.27: Selección de Conjuntos de Entrenamiento aplicada en Iris

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		98.50 %	93.30 %
Cnn	0	92.59 %	92.55 %	74.01 %
Drop1	0	94.52 %	82.92 %	44.66 %
Drop2	0	88.59 %	83.96 %	62.01 %
Drop3	0	94.09 %	85.40 %	62.68 %
Enn	0	4.74 %	99.30 %	94.66 %
Ib2	0	94.81 %	85.65 %	52.01 %
Icf	0	64.07 %	100.00 %	93.31 %
Mcs	0	2.59 %	98.10 %	95.32 %
Multied	0	16.22 %	100.00 %	94.65 %
Renn	0	4.74 %	99.30 %	94.66 %
Rnn	0	91.70 %	99.23 %	82.67 %
Shrink	0	4.74 %	99.30 %	94.66 %
Vsm	0	76.30 %	100.00 %	95.31 %
Ib3	0	89.48 %	94.99 %	87.99 %
Rmhc	7	90.27 %	97.04 %	98.90 %
Ennrs	10	90.27 %	98.25 %	87.98 %
AGG	11	94.81 %	100 %	73.50 %
AGE	11	95.11 %	100 %	68.60 %
CHC	2	96.30 %	74.33 %	53.10 %
PBIL	8	96.52 %	98.72 %	91.30 %

Tabla 2.28: Selección de Conjuntos de Entrenamiento aplicada en Led24Digit

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		84.70 %	62.50 %
Cnn	0	30.61 %	83.15 %	57.52 %
Drop1	0	76.38 %	93.86 %	54.73 %
Drop2	0	22.77 %	83.41 %	60.69 %
Drop3	0	49.75 %	92.19 %	55.89 %
Enn	0	59.51 %	93.54 %	57.49 %
Ib2	0	43.61 %	80.61 %	51.85 %
Icf	0	67.18 %	93.71 %	60.03 %
Mcs	0	31.61 %	89.22 %	59.54 %
Multied	0	93.34 %	100.00 %	16.10 %
Renn	0	66.00 %	98.40 %	59.04 %
Rnn	1	86.22 %	96.21 %	46.26 %
Shrink	0	53.61 %	91.71 %	56.46 %
Vsm	0	62.31 %	87.01 %	47.32 %
Ib3	0	33.99 %	80.05 %	61.24 %
Rmhc	9	90.27 %	82.79 %	48.42 %
Ennrs	14	90.27 %	87.29 %	57.78 %
AGG	13	83.44 %	90.64 %	50.45 %
AGE	13	83.67 %	92.43 %	57.00 %
CHC	10	91.55 %	77.65 %	40.92 %
PBIL	3	87.11 %	91.61 %	52.75 %

Tabla 2.29: Selección de Conjuntos de Entrenamiento aplicada en Led7Digit

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		79.10 %	61.80 %
Cnn	0	36.83 %	75.04 %	54.63 %
Drop1	0	98.69 %	71.44 %	17.76 %
Drop2	0	32.94 %	76.26 %	67.52 %
Drop3	0	88.25 %	77.97 %	18.65 %
Enn	0	79.42 %	96.87 %	29.46 %
Ib2	0	27.50 %	76.69 %	59.80 %
Icf	0	81.46 %	97.01 %	27.07 %
Mcs	0	6.09 %	79.51 %	71.89 %
Multied	0	84.09 %	99.36 %	22.70 %
Renn	0	80.17 %	97.97 %	26.26 %
Rnn	3	96.05 %	68.99 %	34.36 %
Shrink	1	68.48 %	95.88 %	42.47 %
Vsm	0	71.76 %	82.60 %	59.47 %
Ib3	0	64.72 %	57.15 %	50.90 %
Rmhc	51	90.27 %	77.47 %	68.12 %
Ennrs	79	90.27 %	82.69 %	62.84 %
AGG	73	89.58 %	84.34 %	66.30 %
AGE	75	93.69 %	96.53 %	66.15 %
CHC	23	96.93 %	61.61 %	57.37 %
PBIL	45	95.64 %	89.85 %	63.95 %

Tabla 2.30: Selección de Conjuntos de Entrenamiento aplicada en Lymphography

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		81.23 %	29.09 %
Cnn	0	55.99 %	79.50 %	26.93 %
Drop1	0	80.65 %	91.33 %	40.65 %
Drop2	0	20.20 %	77.79 %	37.60 %
Drop3	0	58.35 %	93.19 %	42.05 %
Enn	0	65.16 %	90.31 %	46.97 %
Ib2	0	64.06 %	72.81 %	29.99 %
Icf	0	73.50 %	90.07 %	44.71 %
Mcs	0	41.15 %	82.21 %	40.89 %
Multied	0	88.94 %	100.00 %	45.48 %
Renn	0	70.35 %	95.77 %	46.42 %
Rnn	0	89.79 %	97.19 %	45.48 %
Shrink	0	57.96 %	93.89 %	38.31 %
Vsm	0	68.50 %	80.37 %	46.91 %
Ib3	0	36.72 %	77.23 %	42.99 %
Rmhc	8	90.27 %	75.71 %	46.86 %
Ennrs	12	90.27 %	88.68 %	38.88 %
AGG	11	88.36 %	79.05 %	40.30 %
AGE	12	89.50 %	70.33 %	47.15 %
CHC	7	92.50 %	83.52 %	43.88 %
PBIL	9	91.52 %	89.19 %	38.55 %

Tabla 2.31: Selección de Conjuntos de Entrenamiento aplicada en Monk

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		92.50 %	93.20 %
Cnn	0	90.33 %	77.47 %	37.75 %
Drop1	0	77.91 %	77.82 %	60.48 %
Drop2	0	58.54 %	72.91 %	65.96 %
Drop3	0	70.76 %	66.82 %	56.52 %
Enn	0	29.29 %	91.05 %	76.87 %
Ib2	0	98.23 %	85.48 %	32.90 %
Icf	0	69.26 %	91.10 %	59.95 %
Mcs	0	25.39 %	87.78 %	72.68 %
Multied	0	52.29 %	100.00 %	67.10 %
Renn	0	30.25 %	92.15 %	77.57 %
Rnn	3	88.97 %	81.00 %	67.10 %
Shrink	0	18.70 %	96.57 %	81.72 %
Vsm	0	81.53 %	78.57 %	61.13 %
Ib3	0	67.31 %	63.23 %	65.94 %
Rmhc	79	90.27 %	92.81 %	80.94 %
Ennrs	91	90.27 %	87.59 %	61.58 %
AGG	88	81.48 %	79.39 %	63.35 %
AGE	90	82.90 %	76.03 %	65.55 %
CHC	19	97.74 %	79.09 %	60.91 %
PBIL	28	93.11 %	87.78 %	66.35 %

Tabla 2.32: Selección de Conjuntos de Entrenamiento aplicada en Pima

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		84.70 %	72.70 %
Cnn	0	94.40 %	92.20 %	43.25 %
Drop1	0	85.00 %	91.65 %	70.89 %
Drop2	0	52.07 %	83.39 %	74.46 %
Drop3	0	82.46 %	78.71 %	57.57 %
Enn	0	29.47 %	94.88 %	71.60 %
Ib2	0	93.87 %	84.96 %	43.90 %
Icf	0	77.17 %	92.03 %	69.77 %
Mcs	0	16.45 %	90.21 %	72.25 %
Multied	0	57.74 %	99.97 %	67.43 %
Renn	0	32.57 %	97.06 %	73.03 %
Rnn	5	90.73 %	97.36 %	69.26 %
Shrink	2	24.07 %	95.37 %	73.30 %
Vsm	0	77.56 %	92.05 %	66.91 %
Ib3	0	70.17 %	67.04 %	61.70 %
Rmhc	89	90.27 %	82.12 %	60.70 %
Ennrs	101	90.27 %	95.92 %	70.54 %
AGG	108	80.92 %	94.00 %	70.40 %
AGE	105	89.64 %	94.64 %	70.65 %
CHC	35	98.29 %	95.89 %	69.94 %
PBIL	58	93.06 %	95.70 %	70.40 %

Tabla 2.33: Selección de Conjuntos de Entrenamiento aplicada en Wine

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		98.70 %	88.90 %
Cnn	0	89.51 %	96.38 %	74.39 %
Drop1	0	93.82 %	89.55 %	58.21 %
Drop2	0	90.76 %	94.54 %	69.77 %
Drop3	0	90.78 %	96.01 %	72.95 %
Enn	0	3.93 %	99.29 %	91.06 %
Ib2	0	93.76 %	94.62 %	79.34 %
Icf	0	83.02 %	100.00 %	80.30 %
Mcs	0	1.00 %	98.93 %	90.50 %
Multied	0	17.79 %	99.48 %	90.05 %
Renn	0	3.93 %	99.29 %	91.06 %
Rnn	1	91.01 %	98.12 %	78.76 %
Shrink	0	3.93 %	99.29 %	91.06 %
Vsm	0	80.82 %	97.12 %	71.21 %
Ib3	0	89.26 %	94.23 %	82.63 %
Rmhc	19	90.27 %	99.12 %	93.34 %
Ennrs	23	90.27 %	99.17 %	71.12 %
AGG	25	93.63 %	92.86 %	55.00 %
AGE	27	95.00 %	93.75 %	56.75 %
CHC	9	96.82 %	83.33 %	52.63 %
PBIL	19	94.94 %	98.36 %	90.90 %

Tabla 2.34: Selección de Conjuntos de Entrenamiento aplicada en Wisconsin

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	0		99.00 %	89.90 %
Cnn	0	94.65 %	91.00 %	48.07 %
Drop1	0	98.96 %	87.88 %	80.55 %
Drop2	0	92.37 %	92.67 %	77.74 %
Drop3	0	97.85 %	92.05 %	82.68 %
Enn	0	3.94 %	99.12 %	95.03 %
Ib2	0	97.58 %	98.57 %	72.71 %
Icf	0	95.27 %	97.46 %	92.10 %
Mcs	0	2.33 %	99.18 %	94.30 %
Multied	0	7.40 %	99.79 %	94.44 %
Renn	0	4.13 %	99.12 %	94.88 %
Rnn	4	98.93 %	98.00 %	85.43 %
Shrink	2	2.46 %	99.20 %	94.89 %
Vsm	0	70.70 %	98.95 %	84.63 %
Ib3	0	96.18 %	92.83 %	90.64 %
Rmhc	73	90.27 %	98.05 %	94.18 %
Ennrs	79	90.27 %	99.57 %	94.46 %
AGG	82	89.75 %	99.12 %	93.70 %
AGE	78	98.21 %	91.32 %	90.10 %
CHC	24	99.35 %	87.50 %	68.56 %
PBIL	61	98.36 %	98.74 %	94.80 %

2.D Tablas de Resultados de Conjunto de Datos de Tamaño Mediano en Selección de Conjuntos de Entrenamiento

Tabla 2.35: Selección de Conjuntos de Entrenamiento aplicada en Pen-Based Recognition

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	1		99.27 %	96.46 %
Cnn	4	98.04 %	90.54 %	64.32 %
Drop1	374	98.45 %	92.31 %	65.52 %
Drop2	318	97.69 %	93.28 %	67.89 %
Drop3	391	98.07 %	91.18 %	61.00 %
Enn	269	0.64 %	99.40 %	96.41 %
Ib2	2	98.49 %	88.18 %	56.32 %
Icf	537	92.42 %	97.45 %	73.97 %
Mcs	141	0.36 %	99.28 %	96.26 %
Multied	4274	6.47 %	99.58 %	94.80 %
Renn	579	0.66 %	99.40 %	96.37 %
Rnn	33168	97.52 %	98.42 %	67.09 %
Shrink	277	0.64 %	99.39 %	96.39 %
Vsm	107	0.00 %	99.27 %	96.43 %
Ib3	9	96.42 %	91.74 %	79.63 %
Rmhc	69802	90.02 %	98.19 %	91.40 %
Ennrs	75988	90.02 %	97.71 %	90.40 %
AGG	149281	61.47 %	98.91 %	94.78 %
AGE	103102	61.77 %	98.62 %	94.56 %
CHC	18845	98.99 %	98.51 %	92.33 %
PBIL	83923	73.59 %	99.06 %	94.85 %

Tabla 2.36: Selección de Conjuntos de Entrenamiento aplicada en SatImage

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	1		97.58 %	86.71 %
Cnn	5	95.93 %	91.69 %	54.83 %
Drop1	206	93.66 %	94.39 %	73.08 %
Drop2	183	83.49 %	94.17 %	70.76 %
Drop3	301	93.25 %	95.26 %	67.46 %
Enn	79	9.67 %	98.45 %	86.40 %
Ib2	3	96.75 %	92.26 %	51.63 %
Icf	378	84.53 %	96.96 %	75.23 %
Mcs	92	4.31 %	97.87 %	86.51 %
Multied	836	24.82 %	99.72 %	84.38 %
Renn	490	11.05 %	98.58 %	86.77 %
Rnn	4042	95.15 %	99.07 %	48.90 %
Shrink	184	8.86 %	98.39 %	86.58 %
Vsm	99	0.00 %	97.58 %	86.71 %
Ib3	22	84.66 %	93.31 %	75.41 %
Rmhc	15879	90.02 %	96.63 %	85.30 %
Ennrs	16075	90.02 %	96.66 %	80.00 %
AGG	20245	63.07 %	97.40 %	84.16 %
AGE	21512	63.35 %	97.48 %	85.01 %
CHC	2479	99.06 %	96.75 %	84.28 %
PBIL	6917	72.19 %	97.28 %	84.29 %

Tabla 2.37: Selección de Conjuntos de Entrenamiento aplicada en Thyroid

	Tpo	Red	%Ac Trn	%Ac Test
C4.5	1		99.61 %	99.03 %
Cnn	3	98.00 %	98.80 %	96.95 %
Drop1	182	98.06 %	98.88 %	96.60 %
Drop2	143	87.54 %	98.81 %	98.48 %
Drop3	322	97.44 %	98.97 %	97.52 %
Enn	68	7.15 %	99.92 %	98.67 %
Ib2	2	98.11 %	98.73 %	96.45 %
Icf	244	93.17 %	99.14 %	97.97 %
Mcs	71	3.30 %	99.76 %	98.97 %
Multied	224	10.69 %	100.00 %	92.64 %
Renn	399	7.69 %	99.94 %	98.36 %
Rnn	1841	97.98 %	100.00 %	92.50 %
Shrink	156	5.18 %	99.92 %	98.57 %
Vsm	76	3.12 %	99.66 %	99.06 %
Ib3	94	33.93 %	99.78 %	98.96 %
Rmhc	17895	90.03 %	99.52 %	98.70 %
Ennrs	21342	90.03 %	99.59 %	98.80 %
AGG	28945	63.05 %	99.85 %	98.85 %
AGE	39354	63.60 %	99.05 %	98.09 %
CHC	2891	99.83 %	99.61 %	98.85 %
PBIL	7985	73.61 %	99.78 %	99.02 %

Capítulo 3

Selección Evolutiva Estratificada de Instancias en Conjuntos de Datos de Gran Tamaño Aplicada a Clasificación

Los AAEE pueden ser aplicados de manera eficaz al problema de SPP. En el Capítulo 2 han demostrado un comportamiento destacable en éste ámbito, al ser aplicados sobre conjuntos de datos de tamaño pequeño y mediano.

Cualquier algoritmo se ve afectado al aumentar el tamaño del problema sobre el que se aplica. A este problema lo denominamos problema de Escalado, y está caracterizado por: (i) requerimientos de almacenamiento excesivos, (ii) incremento del tiempo de cálculo, y (iii) disminución de la capacidad de generalización, introduciendo ruido y sobreaprendizaje. A estos efectos no deseados habría que añadirles, en el caso de los AAEE, el que aparece debido al incremento en el tamaño del cromosoma. Cromosomas de mayor tamaño aumentan el consumo de

memoria para su almacenamiento, elevan los tiempos de cálculo y pueden reducir la capacidad de convergencia del algoritmo hacia el óptimo global en un número pequeño de evaluaciones.

Para paliar esta situación se propone la combinación de la estratificación de los conjuntos de entrenamiento con los AAEE, que han demostrado previamente su potencial (ver Subsección 2.5.2). La estratificación, al dividir el conjunto inicial en estratos, disminuirá el tamaño del conjunto sobre el cual se aplica el algoritmo evolutivo, lo cual aporta una solución interesante al problema de escalado.

De este modo, en este capítulo se va a evaluar la combinación de estratificación y el algoritmo CHC sobre conjuntos de datos de mayor tamaño. Nuestra propuesta es comparada con otros algoritmos de SPP no evolutivos siguiendo del mismo modo un modelo estratificado. Para llevarlo a cabo hemos efectuado nuestros experimentos incrementando la complejidad y el tamaño de los conjuntos de datos.

Este capítulo se organiza en las siguientes secciones. En la Sección 3.1, introducimos el problema de Escalado y el efecto que produce en los algoritmos de SPP. La Sección 3.2 describe el enfoque estratificado que se aborda para afrontar problemas de mayor tamaño. La Sección 3.3 estará dedicada a describir la propuesta de combinación de la estrategia de estratificación y selección evolutiva de prototipos. En la Sección 3.4, se muestra la metodología seguida en nuestros experimentos. La Sección 3.5 presenta el estudio experimental desarrollado, describiendo la estructura de las tablas de resultados, los resultados y analizando los mismos. Finalmente, en la Sección 3.6, incluimos las conclusiones a las que nos conduce el presente estudio.

3.1. El Problema de Escalado en Selección de Prototipos

La mayoría de algoritmos de SPP tienen problemas para poder evaluar conjuntos de datos de tamaño elevado. La clasificación empleando la regla básica del vecino más cercano (1-NN [CH67, Wil72b]) presenta una serie de inconvenientes comentados en [WM00a]. Como principales problemas habría que destacar la necesidad de almacenar todos los prototipos de entrenamiento para llevar a cabo la clasificación, lo que introduce requerimientos elevados de memoria. La búsqueda

a través del conjunto de entrenamiento para clasificar una nueva instancia obliga a recorrer todas las muestras de entrenamiento, lo que ralentiza notablemente la clasificación. Estos inconvenientes se ven agudizados conforme crece el conjunto de datos a evaluar [FAV99].

En esta sección estudiaremos el efecto producido por el tamaño de los conjuntos de datos sobre el comportamiento de los algoritmos.

Las principales dificultades que deben afrontar son:

- Eficiencia. La eficiencia de los algoritmos de `SPP` no evolutivos es al menos de $O(n^2)$, siendo `n` el número de instancias en el conjunto de datos. Hay otro conjunto de algoritmos, como por ejemplo `Rnn` [Gat72], `Snn` [RWLI75], `Shrink` [KA87], etc., que presentan eficiencias de orden mayor a $O(n^2)$. Dicha situación convierte a estos últimos en prácticamente ineficaces en problemas de tamaño considerable.
- Recursos. La mayoría de los algoritmos empleados necesitan tener almacenado en memoria el conjunto completo de datos para poder ejecutarse. En caso de que el tamaño del conjunto de datos sea demasiado grande, no podría mantenerse en memoria con lo que sería necesario la utilización de disco como memoria de intercambio. El continuado acceso a disco, con el retardo en la ejecución que supone, afecta negativamente a la eficiencia de los algoritmos.
- Generalización. Los algoritmos se ven afectados en sus capacidades de generalización debido al ruido y el sobreaprendizaje que introducen los conjuntos de datos de gran tamaño [KCH$^+$03].
- Representación. A los `AAEE`, además de los anteriores inconvenientes, habría que añadirles el derivado de la representación que emplean para codificar sus cromosomas. Cuando el tamaño del cromosoma es demasiado grande necesita un mayor número de generaciones para converger. Como efecto adicional, aumenta su coste computacional.

Estos desventajas provocan una considerable degradación del comportamiento y los resultados de los algoritmos cuando nos planteamos su aplicación en bases de datos de gran tamaño. Los algoritmos de `SPP` prototipos evaluados directamente sobre el conjunto completo de gran tamaño pueden ser ineficaces e ineficientes.

Para evitar los inconvenientes asociados al problema de Escalado, proponemos una solución basada en la hibridación de la estrategia estratificada con `AAEE`.

3.2. Estrategia de Estratificación

El objetivo de la estratificación consiste en dividir el conjunto inicial de datos en subconjuntos disjuntos manteniendo la distribución de las clases en todos ellos. Debido a que los prototipos son unidades de información independientes unos de otros, podemos agruparlos en estratos sin pérdida de información.

El número de estratos determinará el tamaño de los mismos. Empleando el número de estratos adecuado podemos reducir significativamente el conjunto de entrenamiento. De esta forma podemos solventar los inconvenientes provocados por el elevado tamaño de los conjuntos de datos al seleccionar sobre cada estrato.

Siguiendo la estrategia estratificada, cada conjunto inicial `D` es dividido en `t` conjuntos disjuntos D_j de igual tamaño $(D_1, D_2, ..., D_t)$.

El problema de Escalado se resuelve al aplicar los algoritmos de `SPP` sobre los conjuntos D_j, los cuales presentan un tamaño `t` veces menor que `D`.

El mecanismo aparece reflejado en la Figura 3.1.

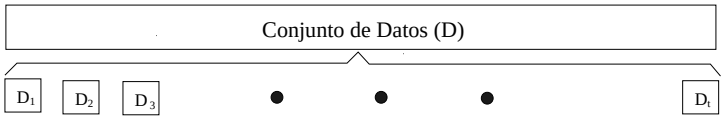

Figura 3.1: Proceso de estratificación

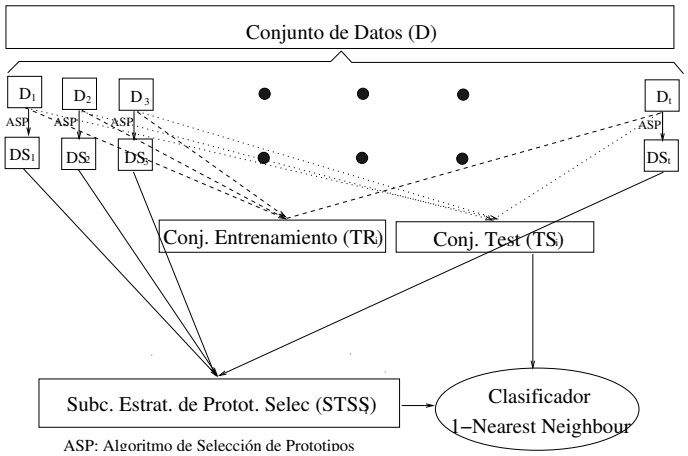

Figura 3.2: Validación cruzada estratificada

3.3. Selección de Prototipos Evolutiva Estratificada

Los algoritmos de SPP siguiendo un modelo estratificado se aplican a cada D_j, con lo que se selecciona un subconjunto al que llamaremos DS_j, siguiendo el modelo que aparece en la Figura 3.2.

El algoritmo evolutivo empleado en la selección es el CHC dado que en el capítulo anterior ha sido el que presenta el mejor compromiso entre eficacia y eficiencia.

El conjunto de test (TS) será el complementario de TR en D, donde los DS_j serán subconjuntos seleccionados a partir de su D_j asociado (ver (3.1) y (3.2)).

$$TR = \bigcup_{j \in J} D_j, J \subset \{1, 2, ..., t\} \tag{3.1}$$

$$TS = D \setminus TR \tag{3.2}$$

El subconjunto seleccionado se obtiene mediante la unión de los DS_j (ver (3.3)) y lo denominaremos STSS. El conjunto STSS es la selección estratificada sobre TR y se obtiene mediante:

$$STSS = \bigcup_{j \in J} DS_j, J \subset \{1, 2, ..., t\} \tag{3.3}$$

La calidad del subconjunto seleccionado `STSS` se evalúa utilizándolo como entrada para el algoritmo 1-vecino más cercano, empleando como test el conjunto `TS`.

En la siguiente sección se describe el modelo de estratos empleado en nuestro estudio.

3.4. Metodología de Experimentación

Los algoritmos han sido evaluados sobre conjuntos de datos de tamaño diferente. En la Subsección 3.4.1 describiremos los conjuntos de datos. La Subsección 3.4.2 describe los algoritmos incluidos en este estudio y sus argumentos, y finalmente, la Subsección 3.4.3 presenta el modelo específico de estratos y particiones seguidos.

3.4.1. Conjuntos de Datos

Para evaluar el comportamiento de los algoritmos sobre diferentes tamaños de conjuntos de datos, hemos efectuado los experimentos aumentando el tamaño y la complejidad de los mismos. Seleccionamos conjuntos de tamaño mediano, grande y muy grande, cuyas características aparecen reflejadas en las Tablas 3.1, 3.2 y 3.3. Estos conjuntos de datos han sido obtenidos del deposito de la `UCI` [MM96]:

Tabla 3.1: Conjuntos de Datos de Tamaño Medio

Conjunto	Instancias	Atributos	Clases
Pen-Based Recognition	10992	16	10
Satimage	6435	36	6
Thyroid	7200	21	3

Estos tres conjuntos han sido previamente descritos en la Sección 2.4.1 del Capítulo 2.

Tabla 3.2: Conjunto de Datos de Tamaño Grande

Conjunto	Instancias	Atributos	Clases
Adult	30132	14	2

La descripción de este conjunto es la siguiente:

Adult: Conjunto de datos extraídos de un censo, donada por Ronny Kohavi y Barry Becker. El objetivo perseguido es determinar si una persona recibe ingresos superiores a cincuenta mil dólares al año.

Tabla 3.3: Conjunto de Datos de Tamaño Muy Grande

Conjunto	Instancias	Atributos	Clases
Kdd Cup'99	494022	41	23

La descripción de este conjunto es la siguiente:

Kdd Cup'99: La tarea que se desea desarrollar consiste en predecir la presencia de intrusos en una red de ordenadores. Se trata de distinguir entre conexiones inadecuadas, llamadas intrusiones o ataques, y conexiones correctas, denominadas normales. El conjunto seleccionado corresponde al subconjunto al 10 % presente en el deposito de la `UCI` [MM96].

3.4.2. Algoritmos y Parámetros

Dividiremos el conjunto de algoritmos evaluados en dos grupos, dependiendo de su naturaleza evolutiva. Para llevar a cabo la selección de los algoritmos empleados en este estudio hemos escogido aquellos más eficientes y eficaces de los presentes en el capítulo anterior.

- Algoritmos no evolutivos. Los algoritmos no evolutivos empleados en este estudio son: `Cnn`, `Drop1`, `Drop2`, `Drop3`, `Ib2` e `Ib3`. Todos ellos están descritos en la Sección 2.2 del Capítulo 2.

- `AAEE`. Se ha seleccionado el algoritmo `CHC` como modelo evolutivo eficiente, basándose en el comportamiento definido en el capítulo anterior (descrito en la Sección 2.3.1 del Capítulo 2).

En la Tabla 3.4 podremos ver los parámetros asociados a cada algoritmo:

Tabla 3.4: Parámetros de los Algoritmos

Algoritmo	Parámetros
Ib3	Aceptabilidad=0.9, Eliminación=0.7
CHC	Población=50, Evaluaciones=10000, α=0.5

3.4.3. Estratificación y Particiones

Cada algoritmo ha sido evaluado siguiendo un proceso de validación cruzada de orden 10. En este proceso de validación, el conjunto de entrenamiento TR_i (`i`=1, ..., 10) es un 90 % de `D` y el de test, TS_i su complementario 10 % de `D`.

Hemos ejecutado los algoritmos de `SPP` desde dos perspectivas en este proceso de validación cruzada:

En primer lugar, los evaluamos del modo que aparece en la Figura 3.3. Denominaremos a esta vía validación cruzada clásica (`Tfcv clásica`) y no es otra que un proceso de validación cruzada de tamaño 10 clásico. La idea es emplear el resultados de esta evaluación para poder compararla con la estratificada.

En `Tfcv clásica` los subconjuntos TR_i y TS_i, `i`=1, ..., 10 se obtienen siguiendo (3.4) y (3.5):

$$\begin{array}{c} TR_i = \bigcup_{j \in J} D_j, \\ J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \end{array} \tag{3.4}$$

$$TS_i = D \setminus TR_i \tag{3.5}$$

en ellas, `t` es el número de estratos, y `b` es el número de estratos agrupados (`b`=`t`/10) para llevar a cabo la validación cruzada de orden 10.

Cada TSS_i se obtiene al aplicar el algoritmo de `SPP` sobre el conjunto TR_i.

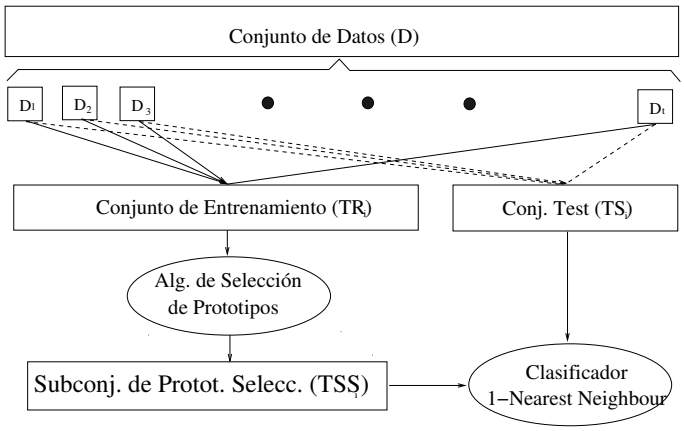

Figura 3.3: Validacion cruzada clásica

La segunda vía a seguir en la validación cruzada es la estratificación, como refleja la Figura 3.2 de la sección anterior. A éste segundo modo de validación la llamaremos validación cruzada estratificada y se denotará como `Tfcv strat`.

En `Tfcv strat`, cada TR_i se define, como podemos ver en (3.4), mediante la unión de subconjuntos D_j (ver Figura 3.2).

En `Tfcv strat`, $STSS_i$ se obtiene mediante la unión de conjuntos DS_j en vez de emplear D_j (ver (3.6)).

$$\begin{gathered} STSS_i = \bigcup_{j \in J} DS_j, \\ J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \end{gathered} \tag{3.6}$$

$STSS_i$ estará compuesto por las instancias seleccionadas por el algoritmo de `SPP` de prototipos en TR_i siguiendo una estrategia estratificada.

El subconjunto TS_i se define mediante la (3.5). Tanto TR_i como TS_i se generan del mismo modo en `Tfcv clásica` y `Tfcv strat`.

En cada conjunto de datos hemos empleado el número de estratos `t` que aparece en las Tablas 3.5 y 3.6.

Tabla 3.5: Estratificación en Conjuntos de Datos Medianos

Pen-Based Recognition	Satimage	Thyroid
t=10	t=10	t=10
t=30	t=30	t=30

Tabla 3.6: Estratificación en Conjuntos de Datos Grandes y Muy Grandes

Adult	Kdd Cup'99
t=10	t=100
t=50	t=200
t=100	t=300

3.5. Estudio Experimental

Esta sección está dedicada a mostrar el estudio experimental llevado a cabo para evaluar el comportamiento de la propuesta de `SPP` evolutiva estratificada. La Subsección 3.5.1 muestra la estructura de la tabla en la que se han almacenado los resultados. En la Subsección 3.5.2 encontramos los resultados y finalmente, en la Subsección 3.5.3 el análisis de los mismos.

3.5.1. Estructura de la Tabla de Resultados

La tabla muestra los resultados obtenidos por los algoritmos tanto evolutivos como no evolutivos. Para reflejar el nivel de robustez de cada algoritmo, los resultados que aparecen son la media del proceso de validación cruzada de orden 10.

Cada columna de la tabla muestra lo siguiente:

- La primera columna muestra el nombre del algoritmo. Cada nombre apa-

recerá acompañado por el tipo de validación empleado, `cl` para indicar `Tfcv clásica` o `st` para `Tfcv strat` (este último acompañado del número de estratos empleado).

- En la segunda columna encontramos el tiempo de ejecución medio (en segundos) de cada algoritmo. Los algoritmos se han ejecutado en un Pentium 4, con 2.4 Ghz, 256 RAM y 40Gb de disco duro.
- La tercera columna contiene el porcentaje de reducción medio conseguido por el algoritmo en la validación cruzada.
- La cuarta columna presenta el porcentaje de acierto en entrenamiento asociado al subconjunto seleccionado. El acierto se calcula empleando la regla del vecino más cercano.
- La quinta columna ofrece el porcentaje de acierto en test del conjunto seleccionado. Al igual que en entrenamiento, el acierto se calcula empleando la regla del vecino más cercano.

3.5.2. Resultados

En todas las tablas de resultados hemos incluido la evaluación del conjunto completo empleando como clasificador la regla del vecino más cercano (`1-NN`). Este resultado lo emplearemos como referencia frente al resto.

Las Tablas 3.7, 3.8 y 3.9 presentan los resultados obtenidos en los conjuntos Pen-Based Recognition, SatImage y Thyroid, respectivamente. Debido a su menor tamaño hemos llevado a cabo las ejecuciones empleando ambas vias de validación cruzada.

En la Tabla 3.10 se muestran los resultados conseguidos tras evaluar el conjunto Adult. Para este conjunto se ha llevado a cabo, en aquellos casos en los que el algoritmo de `SPP` nos lo permite, la evaluación siguiendo `Tfcv clásica` y `Tfcv strat`.

Finalmente, la Tabla 3.11 contiene los resultados asociados a la base de datos de Kdd Cup'99. Este conjunto es el que presenta un mayor número de instancias, atributos y clases. Este hecho provoca que algunos algoritmos (la familia `Drop`) que necesitan una mayor cantidad de recursos para ser ejecutados no puedan ser evaluados.

Tabla 3.7: Resultados para el Conjunto de Datos Pen-Based Recognition

Algoritmo	Tpo	%Red	%Ac Trn	%Ac Tst
1-NN cl	66		99.36	99.39
Cnn cl	4	98.04	84.85	85.69
Cnn st 10	0.20	91.81	93.78	95.43
Cnn st 30	0.07	82.48	97.51	98.63
Drop1 cl	374	98.45	86.23	86.02
Drop1 st 10	2	99.86	57.14	22.00
Drop1 st 30	0.23	99.70	68.96	38.90
Drop2 cl	318	97.69	91.03	91.06
Drop2 st 10	1.9	98.50	52.98	62.92
Drop2 st 30	0.27	95.37	81.83	78.08
Drop3 cl	391	98.07	90.33	90.05
Drop3 st 10	2.1	99.66	53.12	40.91
Drop3 st 30	0.23	98.60	90.51	57.53
Ib2 cl	2	98.49	74.20	75.04
Ib2 st 10	0.1	94.31	93.73	91.41
Ib2 st 30	0.03	88.34	96.25	97.80
Ib3 cl	9	96.42	96.73	98.00
Ib3 st 10	0.2	88.34	92.95	98.44
Ib3 st 30	0.1	83.05	97.07	98.63
CHC cl	18845	98.99	96.29	98.94
CHC st 10	127	96.65	98.85	97.35
CHC st 30	31	93.78	99.69	97.53

Tabla 3.8: Resultados para el Conjunto de Datos SatImage

Algoritmo	Tpo	%Red.	%Ac Trn	%Ac Tst
1-NN cl	36		90.33	90.41
Cnn cl	5	95.93	60.63	61.96
Cnn st 10	0.1	88.42	68.91	75.62
Cnn st 30	0.10	79.49	76.37	80.46
Drop1 cl	206	93.66	84.29	81.68
Drop1 st 10	1.3	98.03	83.18	38.12
Drop1 st 30	0.13	97.89	86.20	30.69
Drop2 cl	183	83.49	83.45	83.51
Drop2 st 10	1.2	83.55	58.21	79.53
Drop2 st 30	0.20	80.85	65.07	79.06
Drop3 cl	301	93.25	87.93	81.03
Drop3 st 10	1.00	96.81	66.46	73.02
Drop3 st 30	0.13	96.65	71.14	57.65
Ib2 cl	3	96.75	59.00	59.59
Ib2 st 10	0.20	91.87	72.15	66.87
Ib2 st 30	0.07	85.77	75.56	75.81
Ib3 cl	22	84.66	84.51	86.45
Ib3 st 10	0.30	78.11	68.95	87.50
Ib3 st 30	0.10	73.71	77.40	87.90
CHC cl	2479	99.06	89.45	89.67
CHC st 10	57	97.52	95.23	88.28
CHC st 30	30	94.32	97.19	89.76

Tabla 3.9: Resultados para el Conjunto de Datos Thyroid

Algoritmo	Tpo	%Rd	%Ac Trn	%Ac Tst
1-NN cl	28		92.87	92.74
Cnn cl	3	98.00	92.50	92.86
Cnn st 10	0.10	90.72	73.13	90.66
Cnn st 30	0.02	84.32	76.47	89.58
Drop1 cl	182	98.06	63.47	62.86
Drop1 st 10	1.00	99.21	80.39	90.25
Drop1 st 30	0.13	99.36	82.22	92.5
Drop2 cl	143	87.54	91.37	91.15
Drop2 st 10	0.70	87.67	53.40	81.19
Drop2 st 30	0.13	86.25	61.94	81.25
Drop3 cl	322	97.44	88.82	85.24
Drop3 st 10	0.80	99.45	80.55	84.81
Drop3 st 30	0.10	99.71	91.17	91.66
Ib2 cl	2	98.11	92.53	92.89
Ib2 st 10	0.10	92.92	76.50	90.80
Ib2 st 30	0.01	85.41	76.58	89.58
Ib3 cl	94	33.93	93.22	93.38
Ib3 st 10	0.50	38.62	93.11	92.33
Ib3 st 30	0.03	33.17	93.70	94.16
CHC cl	2891	99.83	94.20	91.98
CHC st 10	54	99.44	88.25	94.01
CHC st 30	33	99.16	96.49	93.33

Tabla 3.10: Resultados para el Conjunto de Datos Adult

Algoritmo	Tpo	%Rd	%Ac Trn	%Ac Tst
1-NN cl	24		79.34	79.24
Cnn cl	4	99.21	26.40	26.56
Cnn st 10	1	97.34	35.37	32.02
Cnn st 50	0	93.69	66.51	57.42
Cnn st 100	0	90.09	64.42	58.27
Drop1 st 10	44	95.09	100.00	25.64
Drop1 st 50	1	94.59	100.00	24.96
Drop1 st 100	0	94.49	100.00	24.83
Drop2 st 10	48	70.33	27.71	61.30
Drop2 st 50	0	68.03	56.90	70.27
Drop2 st 100	0	66.96	59.31	71.85
Drop3 st 10	41	95.57	48.98	63.46
Drop3 st 50	0	95.34	64.83	71.19
Drop3 st 100	0	93.71	65.82	70.19
Ib2 cl	2	99.94	25.20	25.14
Ib2 st 10	1	99.57	52.33	26.89
Ib2 st 50	0	98.66	74.72	45.68
Ib2 st 100	0	94.33	67.66	54.30
Ib3 cl	210	98.66	74.72	45.68
Ib3 st 10	3	76.69	33.98	70.96
Ib3 st 50	0	73.48	63.93	74.36
Ib3 st 100	0	71.21	68.12	71.52
CHC st 10	20172	99.38	97.02	81.92
CHC st 50	48	98.34	93.66	80.17
CHC st 100	14	97.03	94.28	77.81

Tabla 3.11: Resultados para el Conjunto de Datos Kdd Cup'99

Algoritmo	Tpo	%Rd	%Ac Trn	%Ac Tst
1-NN cl	18568		99.91	99.91
Cnn st 100	8	81.61	99.30	99.27
Cnn st 200	3	65.57	99.90	99.15
Cnn st 300	1	63.38	99.89	98.73
Ib2 st 100	7	82.01	97.90	98.19
Ib2 st 200	3	65.66	99.93	98.71
Ib2 st 300	2	60.31	99.89	99.03
Ib3 st 100	2	78.82	93.83	98.82
Ib3 st 200	0	98.27	98.37	98.93
Ib3 st 300	0	97.97	97.92	99.27
CHC st 100	1960	99.68	99.21	99.43
CHC st 200	418	99.48	99.92	99.23
CHC st 300	208	99.28	99.93	99.19

3.5.3. Análisis de Resultados

A continuación ofrecemos un análisis de los resultados presentes entre las Tablas 3.7 y 3.11 según diferentes criterios:

Eficiencia:

La estratificación reduce significativamente el tiempo de ejecución de los algoritmos, como se puede ver en la segunda columna de las tablas. La reducción conseguida permite la evaluación de algoritmos que necesitan demasiados recursos para poder ser ejecutados de otro modo, o bien reduce sensiblemente su tiempo de ejecución.

En el caso del `CHC` se produce un descenso más que notable en sus tiempos de ejecución. De esta forma el problema de Escalado que aparece en los `AAEE` se ve paliado.

En el conjunto de datos Adult (Tabla 3.10) hay que destacar que aquellos algoritmos que consumen una mayor cantidad de recursos no pueden ser evaluados siguiendo `Tfcv clásica`.

Esta misma situación aparece en la ejecución de los algoritmos sobre el conjunto Kdd Cup'99 (Tabla 3.11). Algunos de los algoritmos no evolutivos no pueden ser aplicados siguiendo los procedimientos de validación comentados. En esta tabla, observando la segunda columna podemos ver el coste asociado a la ejecución del vecino más cercano (`1-NN`) sobre el conjunto completo de datos. Parece clara

la necesidad de llevar a cabo una reducción del conjunto con objeto de poder procesarlo de forma adecuada.

Teniendo como objetivo el tiempo de ejecución podemos destacar lo siguiente:

- La estratificación reduce significativamente el tiempo de ejecución.
- Los algoritmos no evolutivos presentan tiempos de ejecución menores que los evolutivos. A continuación veremos si esa rapidez va acompañada de eficacia.

Porcentaje de Reducción:

Estudiando la tercera columna en cada una de las tablas podremos comprobar el efecto que tiene la estratificación sobre el porcentaje de reducción. Como se puede apreciar, el porcentaje de reducción decrece conforme aumenta el número de estratos. Esto es debido a que al aumentarse el número de estratos, el número de subconjuntos seleccionados a reunir para conformar la solución final es mayor. Existen una serie de muestras importantes (en clasificación) y muy similares que pueden estar en diferentes estratos. Dichas instancias serían seleccionadas en cada estrato, de forma que al final aparecerían en el conjunto final desempeñando la misma tarea en clasificación. Esta situación provoca que el conjunto final seleccionado por un algoritmo siguiendo la estrategia estratificada presente un tamaño ligeramente superior al que se obtendría empleando ese mismo algoritmo sin estratificar.

Los mejores porcentajes de reducción son los ofrecidos por la combinación de estratificación y `CHC`, que mejoran a los no evolutivos en todos los conjuntos estudiados.

Con respecto al porcentaje de reducción considerado como objetivo particular habría que destacar:

- La estratificación afecta ligeramente al porcentaje de reducción, disminuyéndolo conforme aumenta el número de estratos.
- El `CHC estratificado` presenta los mejores porcentajes de reducción.

Porcentaje de Acierto:

Estudiando la última columna de las tablas podemos evaluar el comportamiento en clasificación de cada algoritmo. De este modo podemos ver como los algoritmos no evolutivos (empleando estratificación o no) no mejoran al `1-NN`.

El algoritmo `CHC estratificado` es aquel que presenta un comportamiento más constante en todos los conjuntos, igualando e incluso superando los porcentajes de clasificación obtenidos por 1-`NN` sobre el conjunto completo.

Tomando el porcentaje de acierto como objetivo podemos destacar:

- El algoritmo 1-`NN` aplicado sobre el conjunto completo consigue los mejores porcentajes de acierto en la mayoría de los conjuntos, a costa de mantener íntegro el tamaño del conjunto.
- El algoritmo `CHC estratificado` es aquel que se acerca más y que incluso supera a los porcentajes conseguidos por el 1-`NN`.

Eficiencia frente a Eficacia:

El algoritmo `CHC estratificado` es el que ofrece el mejor equilibrio acierto-reducción. Consigue disminuir el conjunto inicial en un ≈98 % en todos los conjuntos de datos, manteniendo e incluso mejorando los porcentajes de acierto proporcionados por 1-`NN`. El algoritmo `CHC estratificado` es el más eficaz de entre todos los evaluados.

Los algoritmos no evolutivos son más rápidos que el `CHC`, sin embargo presentan porcentajes de acierto y reducción mucho menores. Incrementando el número de estratos reducimos la diferencia en tiempos de ejecución que hay entre ellos (evolutivos y no evolutivos).

El algoritmo `CHC estratificado` en un conjunto de gran tamaño como es el caso de Kdd Cup'99 (Tabla 3.11) presenta los mejores resultados, reduciéndolo en ≈99.5 % (de 494022 a 2470 instancias), con un porcentaje de acierto entorno a 99.2 % y todo ello en 208 segundos por estrato (empleando 300 estratos).

Buscando un equilibrio de objetivos podríamos apuntar:

- Los algoritmos no evolutivos son más eficientes que los evolutivos, pero no son más eficaces.
- El algoritmo `CHC` presenta el mejor equilibrio en reducción, acierto y tiempo de ejecución.

Los algoritmos no evolutivos pueden ejecutarse de forma más eficiente siguiendo una estrategia estratificada, sin embargo no mantienen su eficacia. No presentan un comportamiento equilibrado entre reducción y acierto.

El algoritmo CHC siguiendo una estrategia estratificada mejora a los no evolutivos, ofreciendo los mejores resultados en cuanto a reducción, acierto y recursos necesitados. Es capaz de llevar a cabo reducciones del conjunto inicial de ≈99 % conservando los porcentajes de acierto muy similares a los del 1-NN. Este descenso en el consumo de recursos disminuye el problema de Escalado, mejorando la eficiencia del CHC, conservando su eficacia.

3.6. Comentarios Finales

En este capítulo se ha evaluado la combinación de estratificación y AAEE aplicada a conjuntos de datos de gran tamaño para clasificación. Hemos comparado nuestra propuesta con otros algoritmos de SPP no evolutivos siguiendo un modelo estratificado, aumentando la complejidad y el tamaño de los conjuntos de datos.

Las principales conclusiones alcanzadas son las siguientes:

- La adecuada elección del numero de estratos disminuye significativamente el tiempo de ejecución y el consumo de recursos, manteniéndose el comportamiento del algoritmo con respecto a los porcentajes de acierto y reducción.
- La estrategia de estratificación aplicada a los algoritmos no evolutivos permite reducir la cantidad de recursos que necesitan y mejorar su eficiencia, pero no su eficacia.
- El algoritmo CHC estratificado es el que obtiene los mejores porcentajes de reducción en los conjuntos de datos evaluados. Nuestra propuesta reduce el tamaño del conjunto inicial en más de un 95 % en todos los casos.
- El algoritmo CHC estratificado presenta un comportamiento en clasificación similar al que ofrece el algoritmo 1-NN aplicado sobre el conjunto completo.
- El algoritmo CHC estratificado ofrece los mejores resultados en los conjuntos evaluados, independientemente del tamaño del conjunto sobre el que se aplique (de 7200 instancias en Thyroid hasta 494022 en Kdd Cup'99).

- Nuestra propuesta consigue el mejor equilibrio entre acierto, reducción, tiempo de ejecución y volumen de recursos en todos los conjuntos evaluados, superando a los no evolutivos.

Como conclusión final consideramos nuestra propuesta, en la que combinamos estratificación y `CHC`, como el mejor mecanismo para llevar a cabo la `SPP` sobre conjuntos de datos de gran tamaño. El algoritmo `CHC` selecciona las muestras más representativas, de forma que satisfacen que el subconjunto seleccionado presente elevados porcentajes de acierto y reducción. La estratificación reduce el espacio de búsqueda lo que posibilita el poder evaluar los algoritmos en un tiempo razonable y disminuir la cantidad de recursos requeridos para su ejecución.

Capítulo 4

Selección de Conjuntos de Entrenamiento Evolutiva Estratificada en Conjuntos de Datos de Gran Tamaño para la Generación de Modelos Predictivos y Descriptivos

La extracción de modelos para la toma de decisiones a partir de conjuntos de datos es un proceso básico en MDD [WF00, HMS01, HRF04]. Los modelos, dependiendo del dominio al cual se deseen aplicar, podrán ser o bien predictivos o bien descriptivos. En los modelos predictivos, el objetivo principal es la capacidad de clasificación del modelo así como su interpretabilidad, mientras que en los descriptivos se busca encontrar relaciones o patrones de comportamiento. Un nuevo tipo de modelos descriptivos son los generados para el descubrimiento de subgrupos, donde se pretende obtener modelos descriptivos empleando mecanismos

predictivos [GL02, LKFT04].

El objetivo perseguido en este capítulo se centra en analizar los conjuntos de entrenamiento seleccionados mediante `SPP` para obtener modelos predictivos con capacidades de clasificación e interpretabilidad elevadas por un lado, y que permitan obtener modelos descriptivos de altas prestaciones basándonos en índices propios del descubrimiento de subgrupos por otro.

Para llevar a cabo la selección de los conjuntos de entrenamiento en el Capítulo 2 se mostró que el algoritmo evolutivo `CHC` ofrece muy buen comportamiento en este ámbito de actuación. Debido al tamaño de los conjuntos sobre los que se aplica la `SPP` será necesario emplear la versión estratificada descrita en el capítulo anterior. Los conjuntos de entrenamiento seleccionados se utilizarán para generar modelos de reglas a partir de ellos empleando el algoritmo `C4.5`. La calidad de los conjuntos de entrenamiento se medirá en función de su empleo en el ámbito predictivo o descriptivo. Se analiza así mismo en este capítulo otros algoritmos de extracción de reglas empleando selección evolutiva de conjuntos de entrenamiento, estudiando los modelos predictivo y descriptivo que generan.

El capítulo se organiza en las siguientes secciones. En la Sección 4.1, se describe el contexto del aprendizaje de modelos predictivos y descriptivos, destacando el descubrimiento de subgrupos dentro de los descriptivos. La Sección 4.2 presenta el proceso de selección evolutivo estratificado de conjuntos de entrenamiento. La Sección 4.3 muestra el estudio experimental desarrollado evaluando los algoritmos de selección de conjuntos de entrenamiento de los que extraer modelos predictivos y descriptivos, incluyendo los resultados y el análisis de los mismos. En la Sección 4.4 analizamos diferentes algoritmos de extracción de modelos con respecto al mecanismo de selección evolutiva de conjuntos de entrenamiento estudiado en el apartado anterior, incluyendo los resultados y su análisis. Finalmente, en la Sección 4.5, se alcanzan las conclusiones del capítulo.

4.1. Aprendizaje de Modelos Predictivos y Descriptivos

El aprendizaje de reglas ha sido utilizado comúnmente para la generación de modelos predictivos y descriptivos. Dentro de los descriptivos centraremos la atención en un nuevo modelo basado en el descubrimiento de subgrupos. En esta

sección vamos a estudiar ambas perspectivas, presentando las medidas de calidad empleadas en cada una de ellas para valorar los modelos generados.

En nuestro estudio sobre los algoritmos de selección de conjuntos de entrenamiento la extracción de modelos se llevará a cabo empleando el algoritmo `C4.5`. En la Sección 4.4 analizaremos otros algoritmos de extracción de reglas a partir de los conjuntos seleccionados.

La estructura de esta sección es la siguiente. En la Subsección 4.1.1 analizamos los modelos predictivos y en la 4.1.2 los descriptivos, centrados en los basados en el descubrimiento de subgrupos.

4.1.1. Modelos Predictivos: Reglas de Clasificación

Los modelos predictivos nos sitúan en el contexto del aprendizaje de reglas de clasificación [Coh95]. El aprendizaje de reglas de clasificación tiene como objetivo la inducción predictiva, dedicándose a obtener un conjunto de reglas para ser utilizado en clasificación o predicción.

En el aprendizaje de modelos predictivos se pretende maximizar la precisión del conjunto de reglas inducido, así como aumentar la interpretabilidad del modelo [ZJ03, KMOB04, KME04]. La medidas de calidad que vamos a considerar para valorar los modelos generados serán las que se detallan a continuación.

Porcentaje de Acierto en Test:

Como se ha comentado previamente, a partir del conjunto de entrenamiento seleccionado se genera el modelo mediante el algoritmo de extracción de reglas. Utilizando este modelo se calculará el porcentaje de acierto empleando el conjunto de test asociado al conjunto seleccionado (ver (4.1)).

$$ACTEST = Porcentaje\ de\ Acierto\ en\ Test \tag{4.1}$$

Tamaño del Modelo:

El tamaño del modelo es una medida de la complejidad del mismo. Para evaluar la complejidad sintáctica de las reglas inducidas vamos a estudiar el tamaño del modelo, considerando como tamaño del mismo el número de reglas (n_R) que lo componen (ver (4.2)).

$$TAM = n_R \tag{4.2}$$

Número de Antecedentes:

Como segunda medida del tamaño introduciremos el número medio de antecedentes por regla (ver (4.3) y (4.4)):

$$N_{Antecentes}(R_i) = \sharp|Cond_i| \tag{4.3}$$

donde $N_{Antecedentes}(R_i)$ representa el número de antecedentes de la regla R_i y `ANT` el número de antecedentes medio de las reglas que componen el modelo.

$$ANT = \frac{1}{n_R}\sum_{i=1}^{n_R} N_{Antecedentes}(R_i) \tag{4.4}$$

Ambas medidas (`TAM` y `ANT`) servirán como índices para evaluar la interpretabilidad del modelo.

4.1.2. Modelos Descriptivos: El Descubrimiento de Subgrupos

Los modelos descriptivos habitualmente están destinados al aprendizaje de reglas de asociación [AMS+96]. El aprendizaje de reglas de asociación es un mecanismo de inducción descriptiva que se dedica a descubrir reglas individuales que definen patrones interesantes en los datos.

El descubrimiento de subgrupos como nuevo modelo descriptivo lo podemos definir como la intersección entre la inducción predictiva y descriptiva. Fue formulado inicialmente por Klösgen, con su propuesta de aprendizaje de reglas EXPLORA, y Wrobel con MIDOS [Klö96, Wro01]. En ellos el problema del descubrimiento de subgrupos se define de la siguiente forma:

Dada una población de individuos y una propiedad de esos individuos en la que estamos interesados, buscar subgrupos en esa población que sean estadísticamente "más interesantes", siendo tan grandes como sea posible y ofreciendo el mayor valor de atipicidad estadística con respecto a la propiedad en la que estamos interesados.

En descubrimiento de subgrupos, las reglas son de la forma $Cond \rightarrow Clase$, donde la propiedad de interés es el valor de la clase que aparece en el consecuente de la regla [GL02, LKFT04, LCGF04]. El antecedente de la regla estará compuesto por una conjunción de características (pares atributo-valor) seleccionadas de entre las características que definen las instancias de entrenamiento. Dado que

las reglas se han obtenido a partir de prototipos de entrenamiento etiquetados, el proceso de descubrimiento de grupos se centra en encontrar las propiedades de un conjunto determinado de individuos de la población que satisfacen la propiedad de interés dada. El descubrimiento de subgrupos se pude considerar como un mecanismo de inducción descriptiva dedicándose a encontrar patrones interesantes en los datos. Debido a esta circunstancia, algunas consideraciones estándar llevadas a cabo por los algoritmos de clasificación basados en reglas, tales como el que "las reglas inducidas deben presentar tanta precisión como sea posible" o "las reglas deben ser tan diferentes como sea posible, para cubrir diferentes porciones de la población", deben de ser relajadas.

En descubrimiento de subgrupos el objetivo es encontrar reglas individuales o patrones de interés, los cuales deben ofrecerse en una representación simbólica adecuada de tal forma que puedan ser utilizados con efectividad por potenciales usuarios de esa información. La interpretabilidad de las reglas es por tanto un factor clave en el descubrimiento de subgrupos.

Esta es la razón por la que a menudo se considera diferente el descubrimiento de subgrupos de las tareas propias de clasificación. El descubrimiento de subgrupos se centra en encontrar subgrupos de población interesantes en vez de maximizar la precisión del conjunto de reglas inducido.

Para evaluar el éxito en descubrimiento de subgrupos se estudiarán medidas descriptivas sobre el interés de cada regla obtenida. Las medidas de calidad propuestas consistirán en el valor medio del conjunto de reglas obtenido, lo cual nos permite comparar diferentes algoritmos.

Las medidas de calidad que vamos a considerar en la obtención de estos modelos serán las empleadas por Lavrač et al. en [LKFT04] donde se modifica el algoritmo `CN2` para extraer modelos descriptivos para el descubrimiento de subgrupos (`CN2-SD`). Los índices a considerar serían los siguientes:

Cobertura:

La cobertura mide el porcentaje de ejemplos cubiertos en media por las reglas del conjuntos inducido. La cobertura de un única regla se define como refleja (4.5):

$$Cob(R_i) = Cob(Cond_i \rightarrow Clase) = p(Cond_i) = \frac{n(Cond_i)}{N} \tag{4.5}$$

donde $n(Cond_i)$ es el número de ejemplos cubiertos por la condición $Cond_i$ y N

es el número total de ejemplos. R_i es la i-ésima regla.

La cobertura media para el conjunto de reglas obtenido se calcula como indica (4.6):

$$COB = \frac{1}{n_R} \sum_{i=1}^{n_R} Cob(R_i) \tag{4.6}$$

donde n_R es el número de reglas inducidas.

Confidencia:

La confidencia de una regla nos informa sobre la capacidad de predicción que proporciona. Se obtiene a partir del número de ejemplos positivos de entre todos los que cubre. Se define como se indica en (4.7):

$$Conf(R_i) = Conf(Cond_i \rightarrow Clase) = \frac{n(Cond_i, Clase)}{n(Cond_i)} \tag{4.7}$$

donde $n(Cond_i, Clase)$ es el número de ejemplos cubiertos por $Cond_i$ pertenecientes a $Clase$.

El factor de cobertura media para un conjunto de reglas se calcula como aparece reflejado en (4.8):

$$CONF = \frac{1}{n_R} \sum_{i=1}^{n_R} Conf(R_i) \tag{4.8}$$

Completitud:

En descubrimiento de subgrupos es interesante obtener la completitud global como el porcentaje de ejemplos positivos cubiertos por las reglas. Este índice es calculado considerando la tasa de aciertos positivos para la unión de subgrupos. La completitud de una regla se define como la frecuencia de ejemplos positivos cubiertos (ver (4.9)):

$$Comp(R_i) = Comp(Cond_i \rightarrow Clase) = p(Cond_i, Clase) = \frac{n(Cond_i, Clase)}{N} \tag{4.9}$$

La completitud del conjunto de reglas se calcula según (ver (4.10)):

$$COMP = \frac{1}{N} \sum_{Clase_j} n(Cond_i \bigvee_{Cond_i \rightarrow Clase_j} Clase_j) \tag{4.10}$$

donde los ejemplos cubiertos por varias reglas son enumerados tan solo una vez. Este último parámetro descrito en [LKFT04] en nuestro caso no es relevante empleando la expresión (4.10). La generación de reglas a partir del conjunto seleccionado se lleva a cabo en este capítulo empleando `C4.5`, `CN2` o bien `CN2-SD` [Qui93, CB91, LKFT04]. En el caso de `C4.5`, dicho método genera un modelo tal que cubre todos los ejemplos presentes en el conjunto de entrenamiento al ser un algoritmo de generación por partición, con lo que al valor de `COMP` en todos los casos será de 1. Tanto `CN2` como `CN2-SD` son algoritmos que generan los modelos empleando el método por cobertura, asignándole una clase por defecto a los ejemplos que no se hayan cubierto con las reglas generadas. La situación sería similar a la que aparece con `C4.5` con lo que el índice de completitud vale 1. Al cubrirse todos los ejemplos con los métodos de extracción de reglas empleados, el análisis de este índice no aportaría ningún tipo de información.

Para dotarlo de significado en este estudio, el valor de `COMP` representará la media de la completitud de cada regla. `COMP` se va a encargar de modelizar la capacidad en media que presentan las reglas para cubrir ejemplos. Su nueva definición aparece reflejada en (4.11):

$$COMP = \frac{1}{n_R} \sum_{i=1}^{n_R} Comp(R_i) \tag{4.11}$$

Relevancia:

El grado de relevancia medio de las reglas se calcula empleando la razón de probabilidad de esa regla, normalizada empleando el índice de probabilidad del umbral de importancia (99 %); la media es evaluada empleando todas las reglas. La relevancia refleja lo destacada que es la conclusión alcanzada por las reglas empleando este criterio estadístico.

La relevancia se evalúa empleando el modelo representado en (4.12), similar al utilizado en el algoritmo `CN2` [CN89]:

$$Rel(R_i) = Rel(Cond \rightarrow Clase) = 2 \cdot \sum_j n(Cond, Clase_j) \cdot log \frac{n(Cond, Clase_j)}{n(Clase_j)p(Cond)} \tag{4.12}$$

donde $n(Clase_j)P(Cond)$ es el número esperado de instancias de $Clase_j$ de entre aquellas que satisfacen $Cond$ bajo la hipótesis nula de independencia estadística entre $Clase_j$ y $Cond$. Hay que reseñar que aunque para cada descripción de subgrupo generada se selecciona solo una clase, el criterio de relevancia mide la novedad en la distribución de forma imparcial de cualquier clase, calculándose por tanto la relevancia de la condición de la regla tan solo.

El cálculo de la relevancia media de un conjunto de reglas aparece reflejado en (4.13):

$$REL = \frac{1}{n_R} \sum_{i=1}^{n_R} Rel(R_i) \tag{4.13}$$

Atipicidad:

La atipicidad de una regla se define como la precisión relativa ponderada de una regla, definida como se indica en (4.14) [LFZ99]:

$$Ati(R_i) = Ati(Cond \rightarrow Clase) = p(Cond) \cdot [p(Clase|Cond) - p(Clase)] \tag{4.14}$$

La precisión relativa ponderada puede ser descrita como el equilibrio entre la cobertura de una regla ($p(Cond)$) y su ganancia en precisión ($p(Clase|Cond) - p(Clase)$).

El valor de atipicidad medio de un conjunto de reglas se obtiene como aparece reflejado en (4.15):

$$ATI = \frac{1}{n_R} \sum_{i=1}^{n_R} Ati(R_i) \tag{4.15}$$

Cuanto mayor sea la atipicidad de una regla, más relevante será. Tanto la atipicidad como la relevancia miden la novedad distribucional de un subgrupo, siendo dos de las medidas más importantes en descubrimiento de subgrupos.

Sin embargo, mientras que la relevancia solo tiene en cuenta la novedad en la distribución, la atipicidad también tiene en cuenta la cobertura.

4.2. Selección de Conjuntos de Entrenamiento Evolutiva Estratificada para la Extracción de Modelos Predictivos y Descriptivos

La estrategia seguida para llevar a cabo la selección de conjuntos de entrenamiento consiste en la combinación del modelo estratificado con `AAEE`. Se pretende con ello el poder aplicar la propuesta a conjuntos de cualquier tamaño. Se reduce el espacio de búsqueda mediante la estratificación, al mismo tiempo que el componente evolutivo optimiza la búsqueda en él.

La propuesta presentada es semejante al modelo empleado en el capítulo anterior (ver Sección 3.3). Tanto la representación como la función de fitness son las mismas (ver Secciones 2.3.2 y 2.3.3 del Capítulo 2). La principal diferencia radica en el objetivo perseguido, lo que se ve reflejado en el modo en el cual se evalúa la calidad de los conjuntos finalmente seleccionados.

Del mismo modo que en capítulo anterior, se hace frente al tamaño de los conjuntos a evaluar siguiendo un modelo estratificado. De esta forma, cada conjunto inicial `D` es dividido en `t` conjuntos disjuntos D_j de igual tamaño $(D_1, D_2, ..., D_t)$. Se mantiene la distribución equitativa de las clases en la división del conjunto en estratos.

Los algoritmos de selección de prototipos evolutivos se aplican a cada D_j, con lo que se selecciona un subconjunto al que llamaremos DS_j.

Tras llevar a cabo el proceso de estratificación y selección se reúnen los subconjuntos seleccionados y se procede a la evaluación de la calidad de los mismos empleando el modelo generado a partir de ellos. La Figura 4.1 muestra el proceso.

El conjunto de test (`TS`) será el complementario de `TR` en `D`, donde los DS_j serán subconjuntos seleccionados a partir de su D_j asociado (ver (4.16) y (4.17)). El conjunto `STSS` es la selección estratificada sobre `TR`.

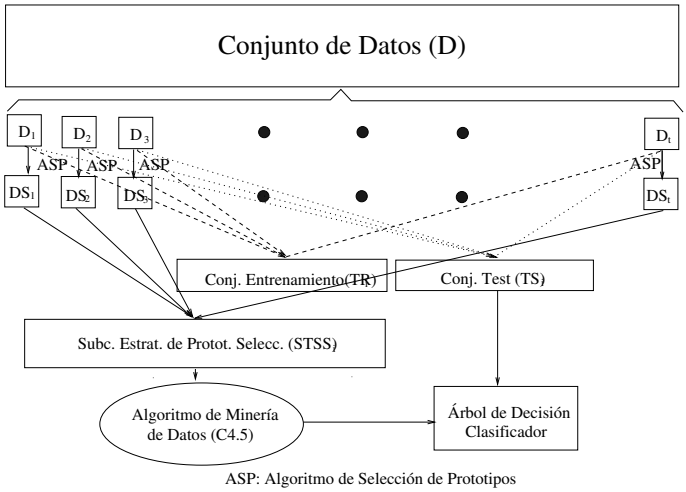

Figura 4.1: Selección de prototipos evolutiva estratificada aplicada a selección de conjuntos de entrenamiento

$$TR = \bigcup_{j \in J} D_j, J \subset \{1, 2, ..., t\} \tag{4.16}$$

$$TS = D \setminus TR \tag{4.17}$$

El subconjunto seleccionado se obtiene mediante la unión de los DS_j (ver (4.18)) y lo denominaremos `STSS`.

$$STSS = \bigcup_{j \in J} DS_j, J \subset \{1, 2, ..., t\} \tag{4.18}$$

En nuestra propuesta, los modelos son extraídos de los subconjuntos seleccionados `STSS` empleando el algoritmo `C4.5`.

4.3. Estudio Experimental de los Algoritmos de Selección de Conjuntos de Entrenamiento para la Extracción de Modelos

Esta sección presenta el estudio experimental desarrollado para analizar las prestaciones de la propuesta ofrecida en selección de conjuntos de entrenamiento, teniendo como objetivos la obtención de modelos predictivos por un lado, y modelos descriptivos para el descubrimiento de subgrupos por otro. Estudiaremos diferentes algoritmos de selección variando el tamaño de los conjuntos de datos, y emplearemos el algoritmo `C4.5` para extraer los modelos de los subconjuntos seleccionados.

La Subsección 4.3.1 muestra la metodología seguida en la experimentación. En la Subsección 4.3.2 se describe la estructura de las tablas que contendrán los resultados. A continuación, en la Subsección 4.3.3, aparecen los resultados y el análisis de los mismos para modelos predictivos. Y finalmente, en la Subsección 4.3.4 se ofrecen los resultados y el análisis con respecto a los modelos descriptivos orientados al descubrimiento de subgrupos.

4.3.1. Metodología de Experimentación

Los algoritmos han sido evaluados sobre conjuntos de tamaño diferente. En este apartado citaremos los conjuntos de datos empleados en la Subsección 4.3.1.1. En la Subsección 4.3.1.2 se presentan los algoritmos y sus correspondientes parámetros, y finalmente en la Subsección 4.3.1.3 el esquema de estratificación y particiones seguido.

4.3.1.1. Conjuntos de Datos

Se han seleccionado conjuntos de datos de tamaño mediano, grande y muy grande según los diferentes análisis efectuados, cuyas características aparecen reflejadas en las Tablas 4.1, 4.2 y 4.3. Estos conjuntos de datos han sido obtenidos del deposito de la `UCI` [MM96]:

Estos conjuntos han sido previamente descritos en las Secciones 2.4.1 y 3.4.1

Tabla 4.1: Conjuntos de Datos de Tamaño Mediano

Conjunto	Instancias	Atributos	Clases
Pen-Based Recognition	10992	16	10
Satimage	6435	36	6
Thyroid	7200	21	3

Tabla 4.2: Conjunto de Datos de Tamaño Grande

Conjunto	Instancias	Atributos	Clases
Adult	30132	14	2

del Capítulo 2 y 3.

4.3.1.2. Algoritmos y Parámetros

Dividiremos el conjunto de algoritmos evaluados en dos grupos, dependiendo de su naturaleza evolutiva. Se han escogido para este estudio aquellos algoritmos más eficientes y eficaces de los presentes en el Capítulo 2.

- Algoritmos no evolutivos. Los algoritmos no evolutivos empleados en este estudio son: `Cnn`, `Drop1`, `Drop2`, `Drop3`, `Ib2` e `Ib3`. Todo ellos descritos en la Sección 2.2 del Capítulo 2.
- Algoritmos evolutivos. Se ha seleccionado el algoritmo `CHC` como algoritmo evolutivo eficaz y eficiente, basándose en el comportamiento definido en los capítulos anteriores (descrito en la Sección 2.3.1 del Capítulo 2).

En la Tabla 4.4 podremos ver los parámetros empleados en cada algoritmo:

Tabla 4.3: Conjunto de Datos de Tamaño Muy Grande

Conjunto	Instancias	Atributos	Clases
Kdd Cup'99	494022	41	23

Tabla 4.4: Parámetros de los Algoritmos de Selección de Conjuntos de Entrenamiento

Algoritmo	Parámetros
Ib3	Aceptabilidad=0.9, Eliminación=0.7
CHC	Población=50, Evaluaciones=10000, α=0.5

4.3.1.3. Estratificación y Particiones

Cada algoritmo ha sido evaluado siguiendo un proceso de validación cruzada de orden 10. En este proceso de validación, el conjunto de entrenamiento TR_i (`i`=1, ..., 10) es un 90 % de `D` y el de test, TS_i su complementario 10 % de `D`.

Hemos ejecutado los algoritmos de selección de prototipos desde dos perspectivas en este proceso de validación cruzada:

En primer lugar, los evaluamos del modo que aparece en la Figura 4.2. Denominaremos a esta vía validación cruzada clásica (`Tfcv clásica`) y no es otra que un proceso de validación cruzada de tamaño 10 clásico. La idea es emplear el resultados de esta evaluación para poder compararla con su versión estratificada.

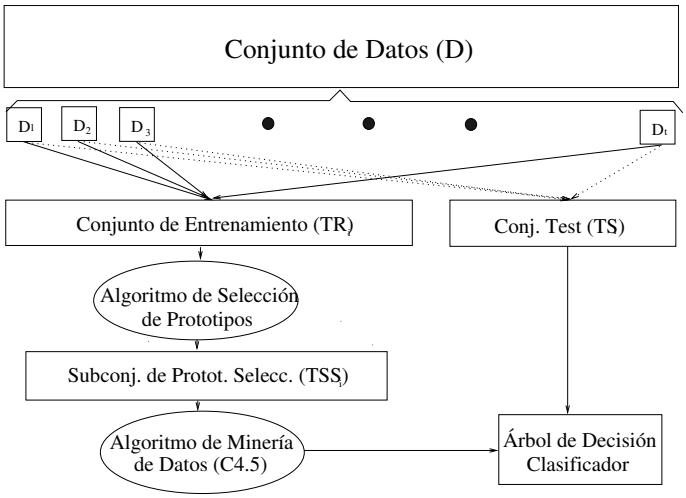

Figura 4.2: Validación cruzada clásica

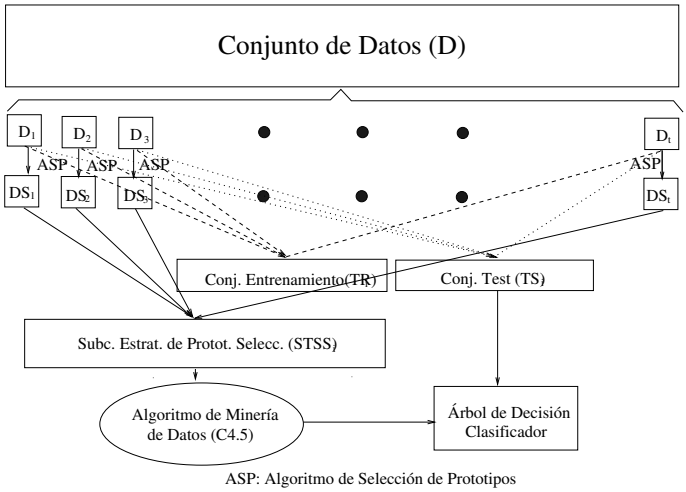

Figura 4.3: Validación cruzada estratificada

En `Tfcv clásica` los subconjuntos TR_i y TS_i, `i`=1, ..., 10 se obtienen siguiendo las expresiones (4.19) y (4.20):

$$\begin{array}{c} TR_i = \bigcup_{j \in J} D_j, \\ J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \end{array} \tag{4.19}$$

$$TS_i = D \setminus TR_i \tag{4.20}$$

en ellas, `t` es el número de estratos, y `b` es el número de estratos agrupados (`b`=`t`/10) para llevar a cabo la validación cruzada de orden 10.

Cada TSS_i se obtiene al aplicar el algoritmo de selección de prototipos sobre el conjunto TR_i.

La segunda vía a seguir en la validación cruzada es la estratificación, como refleja la Figura 4.3. A éste segundo modo de validación la llamaremos validación cruzada estratificada y se denominará `Tfcv strat`.

En `Tfcv strat`, cada TR_i se define, como podemos ver en la (4.19), mediante la unión de subconjuntos D_j (ver Figura 4.3).

El subconjunto TS_i se define mediante (4.20). Tanto TR_i como TS_i se generan del mismo modo en `Tfcv clásica` y `Tfcv strat`.

En `Tfcv strat` $STSS_i$ se obtiene mediante la unión de conjuntos DS_j en vez de emplear D_j (ver (4.21)).

$$\begin{array}{c} STSS_i = \bigcup_{j \in J} DS_j, \\ J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \end{array} \tag{4.21}$$

$STSS_i$ estará compuesto por las instancias seleccionadas por el algoritmo de selección de prototipos en TR_i siguiendo una estrategia estratificada.

En cada conjunto de datos hemos empleado el número de estratos `t` que aparece en la Tabla 4.5:

Tabla 4.5: Estratificación en los Conjuntos de Datos

Pen-Based Recognition	Satimage	Thyroid	Adult	Kdd Cup'99
t=10	t=10	t=10	t=100	t=100

4.3.2. Estructura de las Tablas de Resultados

En esta sección se describe la estructura de las tablas en las que se presentan los resultados. Tendremos dos tipos de tablas: Una donde se muestran los resultados de los conjuntos de entrenamiento seleccionados destinados a obtener modelos predictivos y otra para los conjuntos orientados al descubrimiento de subgrupos.

La tabla asociada a los modelos predictivos presenta la siguiente estructura:

- La primera columna muestra el nombre del algoritmo. Cada nombre aparecerá acompañado por el tipo de validación empleado, `cl` para indicar `Tfcv clásica` o `st` para `Tfcv strat`.
- En la segunda columna encontramos el porcentaje de reducción medio conseguido por el algoritmo de selección de prototipos.
- La tercera ofrece el porcentaje de acierto en test del conjunto seleccionado utilizando el modelo generado por `C4.5`.

- La cuarta columna presenta el número de reglas medio que componen cada modelo.
- En la quinta columna se muestra el número de antecedentes medio que conforman las reglas que componen el modelo.

La segunda tabla está dedicada a presentar los resultados obtenidos desde la perspectiva del descubrimiento de subgrupos:

- Al igual que en la tabla anterior, la primera columna muestra el nombre del algoritmo.
- La segunda contiene la cobertura media de las reglas del modelo.
- La tercera columna presenta el índice de atipicidad medio de las reglas.
- La cuarta está dedicada al índice de relevancia medio de las reglas que componen el modelo.
- La quinta está dedicada a la confidencia media de las reglas que componen cada modelo.
- En la sexta columna encontramos la completitud media de las reglas.

En todas las tablas de resultados hemos incluido la evaluación del conjunto completo empleando el algoritmo `C4.5` sin efectuar ningún tipo de reducción. Este resultado lo emplearemos como referencia frente al resto.

El conjunto de datos Kdd Cup'99 es el que presenta el mayor número de instancias, atributos y clases, circunstancia que provoca el que algunos algoritmos (la familia `Drop`) que necesitan mayor cantidad de recursos para ser ejecutados no puedan ser evaluados.

4.3.3. Resultados y Análisis de los Modelos Predictivos

De la Tabla 4.6 a la 4.8 se presentan los resultados obtenidos en los conjuntos Pen-Based Recognition, SatImage y Thyroid, respectivamente. La Tabla 4.9 muestra los resultados conseguidos tras evaluar el conjunto Adult y finalmente, la Tabla 4.10 contiene los resultados asociados a la base de datos Kdd Cup'99.

Tabla 4.6: Calidad de las Reglas en Pen-Based Recognition en Modelos Predictivos.

	%Red	ACTEST	TAM	ANT
C4.5 cl		96.46	185	9.054
Cnn st	91.81	85.01	62	6.870
Drop1 st	99.86	13.84	4	2.000
Drop2 st	98.50	44.31	30	6.033
Drop3 st	99.66	17.68	6	2.666
Ib2 st	94.31	79.30	48	6.291
Ib3 st	83.05	94.05	88	7.715
CHC st	96.65	80.16	29	5.310

Tabla 4.7: Calidad de las Reglas en SatImage en Modelos Predictivos.

	%Red	ACTEST	TAM	ANT
C4.5 cl		86.71	280	10.810
Cnn st	88.42	70.65	89	9.842
Drop1 st	98.03	36.88	13	4.076
Drop2 st	83.55	70.06	166	11.168
Drop3 st	96.81	59.65	25	6.080
Ib2 st	91.87	62.91	68	10.264
Ib3 st	78.11	86.49	186	10.758
CHC st	94.32	78.83	15	4.400

Tabla 4.8: Calidad de las Reglas en Thyroid en Modelos Predictivos.

	%Red	ACTEST	TAM	ANT
C4.5 cl		99.03	25	6.280
Cnn st	90.72	98.52	13	5.077
Drop1 st	99.21	93.55	3	1.667
Drop2 st	87.67	97.11	7	3.286
Drop3 st	99.45	95.79	3	1.667
Ib2 st	92.92	98.61	9	3.889
Ib3 st	38.62	99.01	22	7.000
CHC st	99.44	93.77	2	1.000

Tabla 4.9: Calidad de las Reglas en Adult en Modelos Predictivos.

	%Red	ACTEST	TAM	ANT
C4.5 cl		85.4	359	14.384
Cnn st	97.34	36.4	21	6.048
Drop1 st	95.09	26.3	3	4.000
Drop2 st	70.33	83.1	196	13.316
Drop3 st	95.57	77.3	78	11.282
Ib2 st	99.57	36.4	12	5.083
Ib3 st	76.69	82.7	179	12.865
CHC st	99.38	82.7	5	2.800

Tabla 4.10: Calidad de las Reglas en Kdd Cup'99 en Modelos Predictivos.

	%Red	ACTEST	TAM	ANT
C4.5 cl		99.9	143	11.780
Cnn st	81.61	96.43	83	11.490
Ib2 st	82.01	95.05	58	10.860
Ib3 st	78.82	96.77	74	11.480
CHC st	99.28	98.41	9	3.560

Estudiando los resultados presentes entre la Tabla 4.6 y la 4.10 podemos extraer las siguientes conclusiones:

Porcentaje de Reducción:

En conjuntos medianos, la familia de algoritmos `Drop` supera ligeramente a nuestra propuesta, ofreciendo mayores porcentajes de reducción.

En conjuntos grandes tan solo el algoritmo `Ib2` mejora lévemente a la selección evolutiva estratificada.

Al evaluar conjuntos de tamaño muy grande, nuestra aproximación al problema de selección de conjuntos de entrenamiento es claramente la vencedora.

Considerando el comportamiento de los algoritmos conforme crece el tamaño del conjunto de datos se destaca el `CHC estratificado` por presentar unas prestaciones uniformes en todos los casos, proporcionando los mejores porcentajes de reducción (superiores al 94 %).

Porcentaje de Acierto:

Al mantener el conjunto de entrenamiento intacto, esto es, sin ningún tipo de reducción, el algoritmo `C4.5` es el que ofrece la mayor precisión en clasificación.

En conjuntos de datos medianos, los algoritmos que seleccionan conjuntos de entrenamiento mayores (el `Ib3` por ejemplo) son los que ofrecen mejores prestaciones.

El comportamiento de los algoritmos sobre conjuntos grandes y muy grandes es prácticamente similar. Presentan mejores resultados aquellos que reducen menos.

La propuesta ofrecida se encuentra entre aquellos que mejores resultados ofrecen en clasificación, sobre todo en conjuntos de tamaño superior, viéndose tan solo superada por métodos que conservan en su mayor parte al tamaño del conjunto original.

Tamaño del modelo:

El tamaño del modelo suele estar ligado al tamaño del conjunto de entrenamiento a partir del cual se genera. Normalmente, cuanto mayor sea dicho conjunto, mayor será el modelo obtenido.

De esta forma, los algoritmos de selección que ofrecen las mayores reducciones serán aquellos que produzcan los modelos de menor tamaño. Este comportamiento aparece sea cual sea el tamaño del conjunto de datos de partida sobre el que se apliquen.

Estudiando el número de reglas que componen los modelos se destacan claramente los generados por `C4.5` sin reducción de ningún tipo, siendo los mayores en todos los casos, lo que los hace menos interpretables.

Dado que el `CHC estratificado` es la técnica que homogéneamente consigue los mejores porcentajes en reducción, también tendrá asociado los modelos con menor número de reglas.

Lógicamente, el menor tamaño de los modelos se verá reflejado en la interpretabilidad que proporciona un modelo. A menor tamaño, mayor interpretabilidad.

Número de Antecedentes:

La situación con respecto al número de antecedentes que presenta una regla es similar al caso anterior. En general, cuanto menor sea el conjunto de entrenamiento del que se parta para generar el modelo, menor será la complejidad de las reglas que se obtienen.

De nuevo, las reglas obtenidas por `C4.5` sin reducción son las que presentan mayor complejidad en media.

Independientemente del tamaño del conjunto, los algoritmos de selección con reglas menos complejas son aquellos con mayores porcentajes de reducción.

Nuestra propuesta, al ser de las que mas reducen, ofrece reglas muy interpretables. Habría que destacar que ofrece reglas con menos antecedentes que otros métodos de selección con porcentajes de reducción mayores. Dicha circunstancia refleja la eficacia del mecanismo de selección del `CHC estratificado` considerando la complejidad de las reglas como objetivo.

4.3.4. Resultados y Análisis de los Modelos Descriptivos para Descubrimiento de Subgrupos

De la Tabla 4.11 a la 4.13 se presentan los resultados obtenidos en los conjuntos Pen-Based Recognition, SatImage y Thyroid, respectivamente.

La Tabla 4.14 muestra los resultados conseguidos tras evaluar el conjunto Adult y finalmente, la Tabla 4.15 contiene los resultados asociados a la base de datos Kdd Cup'99.

Tabla 4.11: Calidad de las Reglas en Pen-Based Recognition en Modelos Descriptivos.

	COB	ATI	REL	CONF	COMP
C4.5 cl	0.005	0.004	242.3	0.945	0.005
Cnn st	0.016	0.012	614.9	0.744	0.013
Drop1 st	0.250	0.013	2275.3	0.133	0.028
Drop2 st	0.033	0.012	956.8	0.518	0.015
Drop3 st	0.166	0.016	2525.0	0.312	0.032
Ib2 st	0.020	0.014	738.5	0.708	0.016
Ib3 st	0.011	0.009	464.5	0.751	0.010
CHC st	0.034	0.023	1093.0	0.740	0.026

Tabla 4.12: Calidad de las Reglas en SatImage en Modelos Descriptivos.

	COB	ATI	REL	CONF	COMP
C4.5 cl	0.003	0.0028	70.1	0.895	0.003
Cnn st	0.011	0.0062	178.2	0.580	0.008
Drop1 st	0.076	0.018	781.7	0.349	0.029
Drop2 st	0.007	0.004	121.2	0.647	0.005
Drop3 st	0.040	0.015	466.5	0.494	0.020
Ib2 st	0.015	0.006	211.4	0.531	0.009
Ib3 st	0.007	0.004	120.9	0.682	0.006
CHC st	0.066	0.037	818.8	0.575	0.050

Tabla 4.13: Calidad de las Reglas en Thyroid en Modelos Descriptivos.

	COB	ATI	REL	CONF	COMP
C4.5 cl	0.040	0.005	156.0	0.941	0.039
Cnn st	0.077	0.010	274.3	0.774	0.075
Drop1 st	0.333	0.032	855.9	0.572	0.311
Drop2 st	0.143	0.019	461.8	0.888	0.138
Drop3 st	0.333	0.039	845.4	0.745	0.320
Ib2 st	0.111	0.015	407.8	0.817	0.109
Ib3 st	0.046	0.006	175.3	0.859	0.045
CHC st	0.500	0.043	1278.7	0.607	0.457

Tabla 4.14: Calidad de las Reglas en Adult en Modelos Descriptivos.

	COB	ATI	REL	CONF	COMP
C4.5 cl	0.004	0.001	49.3	0.744	0.003
Cnn st	0.048	0.008	424.9	0.582	0.024
Drop1 st	0.330	0.014	77.5	0.219	0.082
Drop2 st	0.008	0.001	80.5	0.669	0.005
Drop3 st	0.014	0.003	179.8	0.599	0.011
Ib2 st	0.083	0.010	324.2	0.548	0.041
Ib3 st	0.006	0.001	81.1	0.645	0.004
CHC st	0.200	0.037	1515.3	0.805	0.167

Tabla 4.15: Calidad de las Reglas en Kdd Cup'99 en Modelos Descriptivos.

	COB	ATI	REL	CONF	COMP
C4.5 cl	0.007	0.004	794.0	0.606	0.007
Cnn st	0.013	0.007	1361.0	0.495	0.012
Ib2 st	0.017	0.010	1834.7	0.519	0.016
Ib3 st	0.014	0.008	1521.4	0.479	0.013
CHC st	0.111	0.063	11226.5	0.698	0.108

Estudiando los resultados presentes entre las Tablas 4.11 y la 4.15 podemos extraer las siguientes conclusiones:

Cobertura:

El índice de cobertura empleado nos servirá para analizar el equilibrio entre reglas al cubrir ejemplos. Valores pequeños de este índice reflejan la existencia de reglas que cubren un rango pequeño de ejemplos.

En conjuntos de datos pequeños, los métodos que generan modelos de menor tamaño son aquellos que ofrecen las reglas con mayor índice de completitud. De entre ellos habría que destacar a `Drop1`, `Drop3` y `CHC`.

En conjuntos de tamaño grande, el índice de completitud del `CHC estratificado` es, junto con el de `Drop1`, muy superior al resto. En conjuntos de tamaño muy grande, la selección evolutiva estratificada es la que proporciona el mejor comportamiento.

Las reglas generadas por la propuesta son las que ofrecen el mayor equilibrio al cubrir el conjunto de ejemplos, independientemente del tamaño del conjunto de datos.

Atipicidad:

En el estudio de la atipicidad en los modelos de reglas destaca notablemente el método propuesto frente al resto. El algoritmo `CHC estratificado` presenta el mayor índice que el resto de técnicas, independientemente del tamaño del conjunto de datos.

Dado que este valor es clave, junto con el índice de relevancia, en el descubrimiento de subgrupos, se puede destacar el empleo de la técnica propuesta en este ámbito.

Relevancia:

Estudiando el índice de relevancia en el caso de los conjuntos de tamaño mediano se aprecia como la propuesta ofrece el mejor comportamiento en todos los casos. Es tan solo mejorada en el conjunto de datos Pen-Based Recognition por `Drop1` y `Drop3`, que generan modelos de un tamaño muy inferior al de `CHC`.

Cuando el tamaño del conjunto aumenta, el índice de relevancia mostrado por el `CHC estratificado` supera con creces al resto, triplicando al menos al segundo mejor índice.

Confidencia:

La confidencia de una regla nos determina la calidad en predicción que presenta.

En conjuntos de datos de tamaño mediano aparecen destacados aquellos algoritmos con modelos de mayor tamaño. De esta forma, tanto `C4.5` sin reducción como `Ib3`, con la menor reducción, son los que presentan el mejor índice de confidencia.

Sin embargo, cuando el tamaño del conjunto crece, la situación varía en el caso de nuestra propuesta. Tanto en conjuntos de tamaño grande como muy grande, el `CHC estratificado` ofrece los modelos con mayor índice de confidencia medio. Superior incluso que el proporcionado por el `C4.5` sin reducir.

Completitud:

Con este índice se pretende estudiar la cantidad de ejemplos positivos que cubre una regla de entre todo el conjunto de datos.

El comportamiento que apreciamos en este índice es similar al detectado en el caso de la cobertura. Cuanto menor sea el modelo generado, mayor será su índice de completitud asociado.

De esta forma el `CHC estratificado` vuelve a destacarse frente al resto de métodos, sobre todo cuando el tamaño del conjunto aumenta. En conjuntos de tamaño grande y muy grande dobla al menos el valor del índice del segundo mejor algoritmo.

4.4. Análisis de la Selección Evolutiva de Conjuntos de Entrenamiento con Respecto a los Algoritmos de Extracción de Modelos

En la sección anterior hemos estudiado diferentes algoritmos de selección de conjuntos de entrenamiento, evaluándolos desde la perspectiva de la obtención de modelos predictivos y descriptivos extraídos mediante el algoritmo `C4.5`. De este estudio hemos concluido que la selección evolutiva y estratificada en el caso de conjuntos de gran tamaño es la mejor alternativa.

En esta sección se pretende analizar algunas técnicas de extracción de reglas desde varias perspectivas, comparando sus resultados con los conseguidos con la combinación de la selección evolutiva de instancias y C4.5.

Los diferentes análisis que se desean efectuar son los siguientes:

- **Poda del árbol**. Los modelos generados por los algoritmos de extracción de reglas por partición como el C4.5 se caracterizan por ser completos y consistentes, cubriendo todos los ejemplos presentes en el conjunto de entrenamiento. Dicha circunstancia provoca que al ajustarse demasiado al conjunto de entrenamiento pueda aparecer un mal comportamiento al clasificar nuevos ejemplos [Bra02]. Al mismo tiempo, consigue que el modelo sea sensible ante la presencia de ruido en el conjunto de entrenamiento, lo que puede hacer que el modelo se ajuste a dicho ruido y se degraden sus prestaciones. La manera frecuente de limitar este problema es emplear mecanismos de poda [HCBB02]. El mecanismo de poda, además de mejorar la capacidad de generalización del modelo, reduce su tamaño, lo que aumenta su interpretabilidad [KMOB04, KME04].

  El algoritmo C4.5 utilizado en las evaluaciones implementa un mecanismo de prepoda. En esta sección vamos a extraer modelos con C4.5 empleando poda máxima (lo denominaremos C4.5 Max), mínima (lo llamaremos C4.5 Min) y de nivel medio que es la que implementa por defecto (emplearemos como referencia C4.5).

- **Algoritmos de extracción de reglas alternativos**. En este estudio la extracción de reglas a partir de los subconjuntos seleccionados se ha llevado a cabo utilizando el algoritmo C4.5. Dicho algoritmo como hemos mencionado anteriormente emplea un mecanismo particional para la extracción.

  En esta sección pretendemos analizar el empleo de algoritmos de extracción que empleen el método por cobertura y para ello hemos incluido la evaluación de los algoritmos CN2 y CN2-SD. El análisis del algoritmo CN2-SD nos permite comparar a su vez el comportamiento de la propuesta de selección evolutiva de conjuntos de entrenamiento para la obtención de modelos descriptivos aplicados al descubrimiento de subgrupos con un algoritmo diseñado específicamente para este fin.

  Tanto CN2 como CN2-SD al trabajar con atributos continuos efectúan una etapa previa de discretización de los mismos para poder ser ejecutados. Para comprobar el efecto de la discretización hemos introducido en el análisis

variantes de ambos algoritmos, empleando dos mecanismos de discretización distintos: Por un lado, discretización en intervalos de igual anchura de tamaño 10, y por otro la discretización empleada por el algoritmo `ID3` [Qui86, LHTD02].

- **Combinación de selección evolutiva de conjuntos de entrenamiento y algoritmos de extracción de reglas**. El objetivo perseguido es comprobar las prestaciones que ofrecen las diferentes variantes de extracción de modelos consideradas cuando son aplicadas sobre el subconjunto seleccionado por el algoritmo `CHC`.

Se evaluarán los resultados desde la perspectiva de la generación de modelos predictivos y descriptivos para el descubrimiento de subgrupos.

La sección se organiza de la siguiente forma. La Subsección 4.4.1 muestra la metodología seguida en la experimentación. En la Subsección 4.4.2 se describe la estructura de las tablas que contendrán los resultados. A continuación, en la Subsección 4.4.3, aparecen los resultados y el análisis de los mismos para modelos predictivos. Y finalmente, en la Subsección 4.4.4 se ofrecen los resultados y el análisis con respecto a los modelos descriptivos orientados al descubrimiento de subgrupos.

4.4.1. Metodología de Experimentación

Los algoritmos han sido evaluados sobre conjuntos de tamaño diferente. En la Subsección 4.4.1.1 citaremos los conjuntos de datos empleados. La Subsección 4.4.1.2 presenta los algoritmos y sus correspondientes parámetros, y finalmente la Subsección 4.4.1.3 contiene el esquema de estratificación y particiones seguido.

4.4.1.1. Conjuntos de Datos

Se han seleccionado conjuntos de datos de tamaño pequeño y grande cuyas características aparecen reflejadas en las Tablas 4.16 y 4.17. Estos conjuntos de datos se han obtenido del deposito de la `UCI` [MM96] y han sido descritos en las Secciones 2.4.1 y 3.4.1 del Capítulo 2 y 3.

Tabla 4.16: Conjuntos de Datos de Tamaño Pequeño

Conjunto	Instancias	Atributos	Clases
Pima	768	8	2
Wisconsin	683	9	2

Tabla 4.17: Conjunto de Datos de Tamaño Grande

Conjunto	Instancias	Atributos	Clases
Adult	30132	14	2

4.4.1.2. Algoritmos y Parámetros

Los algoritmos de extracción de reglas que hemos analizado para la generación de los árboles de decisión han sido:

- Métodos de partición: El algoritmo `C4.5`, con su poda maximal, media y minimal.
- Métodos de cobertura: Los algoritmos `CN2` y su versión `CN2-SD` propuesta por Lavrač et al. para el descubrimiento de subgrupos [CN89, GL02, LKFT04].

En la Tabla 4.18 podremos ver los parámetros empleados en cada algoritmo:

Tabla 4.18: Parámetros de los Algoritmos de Extracción de Modelos

Algoritmo	Parámetros
CN2	Estrella=5
CN2-SD	Estrella=5, γ=0.5

La versión de `CN2-SD` evaluada emplea la reducción de pesos multiplicativa de las muestras seleccionadas con el valor de γ indicado en la Tabla 4.18 [GL02].

Como se ha comentado en la introducción de la sección, se evaluarán dos variantes de `CN2` y `CN2-SD`, cada una de ellas con un mecanismo de discretización

distinto: Discretización en anchura (`CN2 Anchura` o `CN2-SD Anchura`) y discretización según el algoritmo `ID3` (`CN2 ID3` o `CN2-SD ID3`) [CGB94, Qui86].

Como referencia, hemos introducido la ejecución de los algoritmos de extracción sobre los conjuntos originales sin reducir. En el caso de `CN2` y `CN2-SD` dicha ejecución solo ha podido llevarse a cabo en los conjuntos pequeños debido a su orden de eficiencia.

4.4.1.3. Estratificación y Particiones

Para la evaluación de los conjuntos de tamaño pequeño, Pima y Wisconisn, se ha seguido un proceso de validación cruzada clásico (`Tfcv cl`). Ambos conjuntos pueden ser evaluados sin necesidad de estratificar.

El conjunto de datos Adult, debido a su tamaño, ha sido procesado empleando el proceso de validación cruzada estratificado (`Tfcv st`). Al presentar un tamaño grande, los algoritmos de extracción de reglas `CN2` y `CN2-SD` presentan problemas de eficiencia empleando el conjunto de datos al completo. La estratificación llevada a cabo en Adult lo divide en 100 estratos (t=100).

Ambos mecanismos de validación aparecen descritos en la Subsección 4.3.1.3.

4.4.2. Estructura de las Tablas de Resultados

En esta sección se describe la estructura de las tablas en las que se presentan los resultados. Tendremos dos tipos de tablas: Una donde se muestran los resultados de los conjuntos de entrenamiento seleccionados destinados a obtener modelos predictivos y otra para los conjuntos orientados al descubrimiento de subgrupos.

La tabla asociada a los modelos predictivos presenta la siguiente estructura:

- La primera columna muestra el nombre del algoritmo. Cada nombre aparecerá acompañado por el tipo de validación empleado, `cl` para indicar `Tfcv clásica` o `st` para `Tfcv strat`.
- En la segunda columna encontramos el porcentaje de reducción medio conseguido por el algoritmo de selección de prototipos.
- La tercera ofrece el porcentaje de acierto en test del conjunto seleccionado.

- La cuarta columna presenta el número de reglas medio que componen cada modelo.
- En la quinta columna se muestra el número de antecedentes medio que conforman las reglas que componen el modelo.
- En la sexta y ultima columna se almacena el tiempo en segundos consumido por el algoritmo de extracción de reglas para generar el modelo.

La estructura de la tabla dedicada a contener los resultados orientados al descubrimiento de subgrupos es exactamente la misma que la segunda tabla descrita en la Subsección 4.3.2.

4.4.3. Resultados y Análisis de los Modelos Predictivos

Las Tablas 4.19, 4.20 y 4.21 muestran los resultados conseguidos tras evaluar el conjunto Pima, Wisconsin y Adult respectivamente.

Estudiando los resultados presentes entre la Tabla 4.19 y la 4.21 podemos extraer las siguientes conclusiones, analizando cada uno de los índices calculados y los objetivos perseguidos:

Porcentaje de Reducción:

Al ser el algoritmo `CHC` el único algoritmo de selección de conjuntos de entrenamiento considerado, no se pueden extraer conclusiones en este sentido al compararlo con el resto.

Tan solo destacar el elevado porcentaje de reducción que proporciona el algoritmo `CHC`, siendo superior al 97 % en los tres conjuntos.

Porcentaje de Acierto:

Al mantener el conjunto de entrenamiento intacto, esto es, sin ningún tipo de reducción, el algoritmo `C4.5` es el que ofrece la mayor precisión en clasificación. El aplicar mecanismos de poda mejora los porcentajes de clasificación del `C4.5`.

El `CN2` y el `CN2-SD` al ser evaluados sobre conjuntos pequeños sin reducción, presentan comportamiento dispar, independientemente del mecanismo de discretización que introduzcan. En Pima ofrecen porcentajes de acierto lejanos al conseguido por `C4.5`, mientras que en Wisconsin se acercan a él. En Adult no pueden ser evaluados sobre el conjunto completo debido a su orden en eficiencia.

Tabla 4.19: Extracción de Modelos Predictivos en Pima.

	%Red	ACTEST	TAM	ANT	TPO (seg)
C4.5 Min cl		70.19	28	5.750	1
C4.5 cl		72.70	15	6.200	1
C4.5 Max cl		79.22	3	1.667	1
CN2 Anchura cl		68.83	45	2.347	806
CN2 ID3 cl		62.33	157	1.303	22049
CN2-SD Anchura cl		63.63	15	7.687	1176
CN2-SD ID3 cl		64.93	14	11.923	11119
CHC+CN2 Anchura cl	98.29	74.02	3	1.333	16
CHC+CN2 ID3 cl	98.29	79.22	3	1.333	9
CHC+CN2-SD Anchura cl	98.29	76.62	2	2.500	8
CHC+CN2-SD ID3 cl	98.29	76.62	2	2.000	5
CHC+C4.5 cl	98.29	77.93	2	1.000	1

El porcentaje de acierto de los modelos extraidos a partir del subconjunto de entrenamiento seleccionado por el algoritmo `CHC` se reduce lévemente al disponer de menor cantidad de muestras para confeccionar el modelo.

Tamaño del Modelo:

De entre las evaluaciones sobre el conjunto completo del algoritmo `C4.5`, cuando mayor sea el nivel de poda, menor será el modelo generado.

Tanto `CN2` como `CN2-SD` generan modelos con tamaños similares o superiores a los producidos por el algoritmo `C4.5`. El número de reglas que los conforman aumenta cuando se emplea el mecanismo discretización del algoritmo `ID3`. `CN2` genera un número de reglas superior al `CN2-SD`.

Los modelos extraidos a partir de los subconjuntos seleccionados por el algoritmo `CHC` son los que están compuestos por un menor número de reglas. Al generarse los modelos utilizando un conjunto de entrenamiento de menor tamaño,

Tabla 4.20: Extracción de Modelos Predictivos en Wisconsin.

	%Red	ACTEST	TAM	ANT	TPO (seg)
C4.5 Min cl		91.32	21	5.000	1
C4.5 cl		95.04	12	4.083	1
C4.5 Max cl		92.81	8	3.000	1
CN2 Anchura cl		92.75	15	2.187	357
CN2 ID3 cl		94.20	14	2.400	379
CN2-SD Anchura cl		94.20	9	5.333	674
CN2-SD ID3 cl		94.20	9	5.333	735
CHC+CN2 Anchura cl	99.35	91.30	2	1.000	26
CHC+CN2 ID3 cl	99.35	88.40	2	1.000	2
CHC+CN2-SD Anchura cl	99.35	91.30	2	1.000	12
CHC+CN2-SD ID3 cl	99.35	88.40	4	1.000	5
CHC+C4.5 cl	99.35	88.40	2	1.000	1

el conjunto de reglas necesario para cubrirlos es más pequeño.

Lógicamente, el menor tamaño de los modelos se verá reflejado en la interpretabilidad que proporciona un modelo. A menor tamaño, mayor interpretabilidad.

Número de Antecedentes:

La situación con respecto al número de antecedentes que presenta una regla es similar al caso anterior.

Habría que destacar al algoritmo `CN2-SD` por producir reglas con una gran cantidad de antecedentes, independientemente del mecanismo de discretización empleado.

Los modelos extraidos empleando como entrenamiento a los conjuntos seleccionados por el algoritmo `CHC` son los que ofrecen menor cantidad de antecedentes en sus reglas. Se destaca notablemente la reducción en tamaño conseguida en Adult, donde la combinación de `CHC estratificado` y `C4.5` consigue los modelos más

Tabla 4.21: Extracción de Modelos Predictivos en Adult.

	%Red	ACTEST	TAM	ANT	TPO (seg)
C4.5 Min cl		82.45	1460	15.437	2
C4.5 cl		85.40	359	14.384	2
C4.5 Max cl		85.26	64	11.187	2
CHC+CN2 Anchura st	97.03	75.90	31	2.290	2073
CHC+CN2 ID3 st	97.03	77.22	44	2.000	8318
CHC+CN2-SD Anchura st	97.03	82.17	11	5.272	3325
CHC+CN2-SD ID3 st	97.03	82.83	10	7.000	10029
CHC+C4.5 st	97.03	82.70	5	2.800	1

pequeños. De 359 reglas y 14.384 antecedentes por regla asociados a los modelos generados por el `C4.5` sobre el conjunto Adult completo, se obtienen 5 reglas con 2.8 antecedentes con `CHC estratificado` y `C4.5`.

La combinación de `CHC` y `C4.5` genera los modelos con menor número de reglas y antecedentes, y por tanto, más interpretables.

Tiempos de Ejecución:

En este caso analizamos los tiempos de ejecución de los algoritmos de extracción de reglas a partir del conjunto de entrenamiento.

Lógicamente, la extracción de los modelos sobre el conjunto completo va a ser mas costosa en tiempo que sobre el subconjunto seleccionado por `CHC`.

De entre los evaluados, se destaca notablemente el algoritmo `C4.5`, que genera los modelos en un tiempo no superior a 2 segundos.

Tanto `CN2` como `CN2-SD` presentan tiempos bastante superiores a los del algoritmo `C4.5`. Sobre el conjunto Adult al completo no pueden ser ejecutados debido a su eficiencia. Hecho que queda reflejado al evaluarlos sobre el subconjunto de Adult seleccionado por `CHC`. Al ejecutarse sobre Adult, reducido en un 97.03 %, `CN2` en el mejor caso tarda 2073 segundos, `CN2-SD` consume 3325 segundos, mientras que `C4.5` lo haría en 2 segundos.

Poda del Árbol:

Los modelos generados por el algoritmo `C4.5` con poda mínima produce los modelos mayores con el sobreajuste que esto supone. Al efectuar poda máxima, el tamaño del modelo se reduce considerablemente, así como el porcentaje de acierto al disminuirse la capacidad de generalización del modelo. La mejor opción consiste en llevar a cabo una poda equilibrada, determinada por un parámetro. El inconveniente que presenta esta elección es la elección del valor de ese parámetro, dado que variará dependiendo del conjunto de datos sobre el que se aplique.

Algoritmos de Extracción de Reglas:

El algoritmo `C4.5` en sus versiones con poda es el que consigue los modelos de menor tamaño y mayores porcentajes de acierto.

De entre los métodos de cobertura, el `CN2-SD` consigue con menos reglas que los del `CN2`, pero este a su vez obtiene menos antecedentes por regla. Los porcentajes de acierto no varían sensiblemente, siendo mejores los del `CN2-SD`.

Selección Evolutiva de Conjuntos de Entrenamiento y Algoritmos de Extracción de Reglas:

La combinación de `CHC` y algoritmos de extracción de reglas genera modelos con porcentajes de clasificación cercanos y en algún caso superiores a los que se consiguen sin reducción. La combinación que consigue los modelos más interpretables con mayor precisión es el algoritmo `CHC` con `C4.5`.

El resto de algoritmos de extracción mejoran su comportamiento al aplicarse sobre los conjuntos reducidos seleccionados por `CHC`.

La combinación de `CHC` y `C4.5` consigue modelos con capacidad de predicción elevada, ofrece al mismo tiempo los modelos más interpretables y la extracción de los mismos la lleva a cabo de la forma más eficiente.

4.4.4. Resultados y Análisis de los Modelos Descriptivos para Descubrimiento de Subgrupos

Las Tablas 4.22, 4.23 y 4.24 muestran los resultados conseguidos tras evaluar el conjunto Pima, Wisconsin y Adult respectivamente.

Estudiando los resultados presentes entre la Tabla 4.22 y la 4.24 podemos extraer las siguientes conclusiones:

Cobertura:

Tabla 4.22: Extracción de Modelos Descriptivos en Pima.

	COB	ATI	REL	CONF	COMP
C4.5 Min cl	0.033	0.066	51.799	0.740	0.256
C4.5 cl	0.066	0.019	20.193	0.899	0.054
C4.5 Max cl	0.035	0.010	15.091	0.916	0.030
CN2 Anchura cl	0.012	-0.007	10.029	1.000	0.012
CN2 ID3 cl	0.005	-0.003	4.730	1.000	0.005
CN2-SD Anchura cl	0.434	-0.087	50.099	0.708	0.305
CN2-SD ID3 cl	0.398	-0.080	80.744	0.777	0.311
CHC+CN2 Anchura cl	0.360	-0.160	47.102	0.536	0.202
CHC+CN2 ID3 cl	0.333	-0.184	36.123	0.469	0.244
CHC+CN2-SD Anchura cl	0.434	-0.090	38.686	0.659	0.265
CHC+CN2-SD ID3 cl	0.840	-0.040	18.940	0.565	0.440
CHC+C4.5 cl	0.500	0.088	54.184	0.704	0.366

El índice de cobertura empleado nos servirá para analizar el equilibrio entre reglas al cubrir ejemplos. Valores pequeños de este índice reflejan la existencia de reglas que cubren un rango pequeño de ejemplos.

Se destaca de entre todos los algoritmos de extracción de reglas el `CN2-SD`, que debido al mecanismo de generación de reglas que emplea es el que mayor índice de cobertura presenta. En su caso la búsqueda de reglas no va dirigida únicamente por la precisión de las mismas, si no que emplea una heurística cuyo objetivo es el descubrimiento de subgrupos, lo cual le permite obtener reglas más generales que cubran un mayor número de ejemplos a costa de perder algo de precisión.

En general, las evaluaciones que consiguen modelos con pocas reglas son las que ofrecen índices de cobertura mayores. Así aparecen destacadas las combinaciones del algoritmo `CHC` con cualquiera de los métodos de extracción de reglas.

Tabla 4.23: Extracción de Modelos Descriptivos en Wisconsin.

	COB	ATI	REL	CONF	COMP
C4.5 Min cl	0.047	0.021	36.772	0.993	0.047
C4.5 cl	0.083	0.036	59.804	0.991	0.081
C4.5 Max cl	0.142	0.062	97.207	0.963	0.139
CN2 Anchura cl	0.061	-0.033	49.006	1.000	0.061
CN2 ID3 cl	0.065	-0.035	52.272	1.000	0.065
CN2-SD Anchura cl	0.538	-0.014	302.350	0.964	0.524
CN2-SD ID3 cl	0.535	-0.015	300.226	0.964	0.521
CHC+CN2 Anchura cl	0.500	-0.263	214.551	0.882	0.448
CHC+CN2 ID3 cl	0.529	-0.280	193.873	0.865	0.462
CHC+CN2-SD Anchura cl	0.500	-0.028	214.551	0.882	0.448
CHC+CN2-SD ID3 cl	0.500	-0.017	81.834	0.513	0.304
CHC+C4.5 cl	0.500	0.143	146.295	0.867	0.428

Atipicidad:

En la extracción de modelos descriptivos para el descubrimiento de subgrupos, tanto la atipicidad como el índice de relevancia son factores importantes para valorar la calidad del modelo. Mientras que la relevancia solo tiene en cuenta la novedad en la distribución, la atipicidad también tiene en cuenta la cobertura, convirtiéndose en un índice clave en el descubrimiento de subgrupos.

Las distintas evaluaciones del algoritmo `C4.5` sobre el conjunto original se destacan frente al resto por presentar los mayores valores de éste índice. Concretamente, la combinación del `CHC` y `C4.5` ofrece el índice de mayor atipicidad en todos los conjuntos evaluados, siendo este muy superior al resto.

Relevancia:

Estudiando la relevancia se aprecia que las evaluaciones en las que interviene el algoritmo de extracción de reglas `CN2-SD` son las que presentan un valor más

Tabla 4.24: Extracción de Modelos Descriptivos en Adult.

	COB	ATI	REL	CONF	COMP
C4.5 Min cl	0.002	0.0004	26.219	0.752	0.002
C4.5 cl	0.004	0.001	49.300	0.744	0.003
C4.5 Max cl	0.016	0.003	198.137	0.746	0.013
CHC+CN2 Anchura st	0.026	-0.016	343.541	0.273	0.021
CHC+CN2 ID3 st	0.017	-0.011	212.923	0.194	0.015
CHC+CN2-SD Anchura st	0.408	-0.085	1997.301	0.748	0.298
CHC+CN2-SD ID3 st	0.415	-0.080	2050.153	0.701	0.321
CHC+C4.5 st	0.200	0.037	1515.318	0.805	0.167

elevado de este índice. El algoritmo `CN2-SD` emplea una heurística de extracción de reglas que permite obtener valores elevados de éste índice.

La combinación de `CHC` y `CN2-SD` o `C4.5` obtienen índices de relevancia elevados, sobre todo cuando el tamaño del conjunto sobre el que se aplican es elevado. En el caso de `CN2-SD` debido a su orden de eficiencia, la selección evolutiva estratificada permite aplicarlo sobre conjuntos de tamaño elevado de forma eficaz.

Confidencia:

La confidencia de una regla nos determina la calidad en predicción que presenta.

En los conjuntos evaluados aparecen destacados aquellas ejecuciones que generan modelos de mayor tamaño. Al tener una mayor cantidad de reglas, estas pueden ser mas específicas y tener una menor tasa de error.

De entre las diferentes combinaciones de `CHC` y un algoritmo de extracción de reglas, es el `C4.5` el que presenta mayores índices de confidencia en los conjuntos de diferente tamaño considerados. Esta situación es aún más destacada en el caso de conjuntos de tamaño grande como Adult, donde la combinación mencionada presenta el mejor comportamiento.

Completitud:

Con este índice se pretende estudiar la cantidad de ejemplos positivos que cubre una regla de entre todo el conjunto de datos. Cuanto menor sea el modelo generado, mayor suele ser índice de completitud asociado.

De esta forma el `CHC estratificado` vuelve a destacarse frente al resto de métodos, sobre todo cuando el tamaño del conjunto aumenta.

En este caso no hay ninguna de los algoritmos de extracción que obtenga unos resultados significativamente mejores al resto. El algoritmo `CN2-SD` discretizando según el algoritmo `ID3` combinado con el `CHC` es el que presenta el mejor comportamiento en este sentido.

Poda del Árbol:

Los modelos generados por el `C4.5` sobre el conjunto al completo, independientemente del grado de poda, presentan los índices más discretos en descubrimiento de subgrupos. Considerando la atipicidad como factor clave y la relevancia como importante, se destaca la evaluación de `C4.5` con poda mínima.

Algoritmos de Extracción de Reglas:

El algoritmo `CN2-SD` siguiendo ambas vías de discretización es el que consigue los mayores índices sobre los conjuntos de entrenamiento sin reducción, circunstancia lógica dado que la heurística que sigue busca optimizar su comportamiento en descubrimiento de subgrupos.

Selección Evolutiva de Conjuntos de Entrenamiento y Algoritmos de Extracción de Reglas:

La combinación de `CHC` y `C4.5` se destaca frente al resto ofreciendo los índices más altos, superando incluso al `CN2-SD` en muchos de ellos, sobre todo cuando el tamaño del conjunto se eleva y es necesario estratificar.

De entre los métodos de extracción de reglas por cobertura se destaca el `CN2-SD` con discretización en Anchura, con un comportamiento más homogéneo en los conjuntos evaluados.

Dado que la combinación de `CHC` y `C4.5` ofrece el mejor o uno de los mejores comportamientos en todos los índices, podemos destacarlo como una propuesta adecuada para la extracción de modelos descriptivos destinados al descubrimiento de subgrupos mediante la selección de conjuntos de entrenamiento.

4.5. Comentarios Finales

En este capítulo se ha analizado la propuesta de selección evolutiva estratificada de prototipos aplicada a la selección de conjuntos de entrenamiento desde dos perspectivas: la obtención de modelos predictivos y modelos descriptivos para el descubrimiento de subgrupos. Para ello, se ha comparado la propuesta con otras técnicas de selección de prototipos siguiendo el modelo estratificado, aumentando la complejidad y tamaño de los conjuntos de datos. Así mismo, hemos estudiado diferentes técnicas de extracción de modelos comparando sus resultados con los de la propuesta ofrecida.

Las principales conclusiones alcanzadas en modelos predictivos son las siguientes:

- Considerando la reducción ofrecida por cada algoritmo, la propuesta descrita en este capítulo presenta el mejor comportamiento independientemente del tamaño del conjunto sobre el que se la aplique.
- A nivel del porcentaje de acierto, se destacan frente al resto aquellas técnicas que reducen poco o nada el conjunto de datos. El algoritmo `CHC estratificado` con `C4.5` se encuentra entre los mejores, más aún cuando se aumenta el tamaño del conjunto.
- La combinación de `CHC` y `C4.5` consigue generar los modelos con menor número de reglas y antecedentes por regla, con el consiguiente incremento en la interpretabilidad de los modelos.

Las principales conclusiones alcanzadas en modelos descriptivos aplicados al descubrimiento de subgrupos son las siguientes:

- Tanto a nivel de cobertura como de completitud, el algoritmo propuesto vuelve a destacarse frente al resto. Siendo esta mejora aún más palpable conforme el tamaño del conjunto de datos aumenta.
- En descubrimiento de subgrupos, tanto la atipicidad como la relevancia del conjunto de reglas son factores clave. En ambos aspectos, el algoritmo `CHC estratificado` combinado con `C4.5` supera de forma más que notable al resto de métodos.

Como conclusión final, consideramos la propuesta ofrecida en este capítulo como el mejor mecanismo para seleccionar conjuntos de calidad teniendo como objetivos la generación de modelos predictivos y el descubrimiento de subgrupos. La propuesta es capaz de seleccionar el conjunto de prototipos más representativo de tal forma que los modelos de reglas extraídos a partir de él ofrecen los mayores índices de calidad en ambos casos.

Comentarios Finales

Dedicamos esta sección a resumir brevemente los resultados obtenidos y a destacar las conclusiones que esta memoria aporta. Se incluyen algunos aspectos sobre trabajos futuros que siguen la línea aquí expuesta y sobre otras líneas de investigación que se pueden derivar.

A. Resumen y Conclusiones

Hemos estudiado diferentes `AAEE` para llevar a cabo la `SII` en reducción de datos, teniendo como objetivos el clasificar basándose en el vecino más cercano por un lado y la selección de conjuntos de entrenamiento por otro. Se pretende llevar a cabo la reducción del conjunto inicial sin que las prestaciones de los modelos generados a partir de ellos se vean especialmente afectadas. Para ello, se ha comparado el comportamiento de las técnicas evolutivas con otras no evolutivas. Así mismo, se ha estudiado el problema de escalabilidad que presentan los algoritmos de `SII`, proponiéndose una versión estratificada de los mismos para abordar este problema. Finalmente se analizan los modelos predictivos y descriptivos generados empleando medidas de calidad para estudiar sus prestaciones.

Los siguientes apartados resumen brevemente los resultados obtenidos y presentan algunas conclusiones sobre los mismos. Aquellos resultados que ya han sido publicados van acompañados de las referencias bibliográficas que corresponden.

A.1 Selección Evolutiva de Prototipos en Reducción de Datos

La selección evolutiva de instancias en conjuntos de tamaño pequeño y mediano la hemos aplicado siguiendo dos perspectivas:

- La obtención de un clasificador basado en el vecino más cercano mediante la selección de las muestras de entrenamiento, teniendo como objetivo mejorar su precisión.
- Generar modelos predictivos basados en árboles de decisión a partir de la selección de conjuntos de entrenamiento, analizando la precisión conseguida.

En ambos casos, los resultados presentados siguiendo ambas vías han sido satisfactorios. Como se ha podido comprobar, esto se debe a las siguientes razones:

- Los `AAEE` mejoran el comportamiento de los algoritmos no evolutivos al ofrecer el mayor equilibrio entre precisión y reducción. Consiguen los subconjuntos de menor tamaño con mayores prestaciones. De entre los `AAEE`, cabe destacar el algoritmo `CHC` como el mejor del estudio, tanto en clasificación como en selección de conjuntos de entrenamiento.
- Cuando el tamaño del conjunto de entrada es mayor los algoritmos ven mermadas sus prestaciones, siendo los `AAEE` y en especial el algoritmo `CHC`, los que presentan un comportamiento más estable.

Los resultados obtenidos muestran que se pueden seleccionar subconjuntos de menor tamaño y alta calidad empleando `AAEE`, gracias a sus capacidades de búsqueda sin heurística a priori [CHL02, CHL03b, CHL04b].

A.2 Selección Evolutiva de Prototipos Estratificada en Conjuntos de Datos de Gran Tamaño Aplicada a Clasificación mediante el Vecino Más Cercano

Como ya comenzó a apuntarse en el Capítulo 2 al aplicar los algoritmos de `SII` sobre conjuntos de tamaño mediano, todas las técnicas de `SII` se ven afectadas frente al incremento de tamaño. Dado que la extracción de conocimiento normalmente se llevará a cabo sobre conjuntos con volúmenes considerables, la sensibilidad ante este factor es clave.

La alternativa ofrecida siguiendo la estrategia estratificada permite reducir el dominio de búsqueda a través de su división en estratos y reuniendo posteriormente los resultados de estos. Al estratificar, solventamos el problema del tamaño,

así como al aplicar selección evolutiva utilizamos el mecanismo de selección más efectivo.

Analizando los resultados obtenidos podemos destacar lo siguiente:

- La adecuada elección del número de estratos reduce significativamente el tiempo de ejecución y el consumo de recursos, conservándose el comportamiento de los algoritmos de `SII` con respecto a porcentajes de acierto y reducción.
- La estratificación en algoritmos no evolutivos permite reducir su consumo de recursos y mejorar su eficiencia, pero no su eficacia.
- La propuesta de selección evolutiva estratificada de instancias ofrece el mejor equilibrio entre precisión, reducción, tiempo de ejecución y volumen de recursos en todos los conjuntos evaluados, superando a las técnicas no evolutivas.

La mejora en los resultados conseguidos indica que el uso combinado de selección evolutiva y estratificación presenta un gran potencial en conjuntos de datos de tamaño elevado [CHL03a, CHL04a, CHL04c].

A.3 Selección de Conjuntos de Entrenamiento Evolutiva Estratificada para Obtener Modelos Predictivos y Descriptivos Basados en el Descubrimiento de Subgrupos

En el Capítulo 4 de esta memoria se emplea la hibridación de estratificación y selección evolutiva de instancias para obtener subconjuntos de entrenamiento en conjuntos de gran tamaño. El objetivo es extraer modelos predictivos y modelos descriptivos dedicados al descubrimiento de subgrupos. Se estudian las prestaciones de los modelos predictivos considerando su precisión e interpretabilidad. Las prestaciones de los modelos descriptivos consideran factores tales como su aticidad, relevancia, etc.

Analizando los resultados conseguidos podemos destacar lo siguiente:

- Considerando el tamaño del modelo obtenido, la propuesta de selección evolutiva estratificada presenta el mejor comportamiento independientemente

del tamaño del conjunto sobre el que se aplique. Este método consigue los modelos con el menor tamaño, tanto en número de reglas como en el número de antecedentes por regla, lo cual ofrece mayor interpretabilidad.

- Para analizar la calidad en clasificación de los conjuntos seleccionados se evalúa la precisión obtenida por el modelo generado a partir de ellos utilizando `C4.5`. Se destacan frente al resto aquellas técnicas que reducen poco o nada el conjunto inicial de datos. El algoritmo propuesto se encuentra entre los mejores, más aún cuando se incrementa el tamaño del conjunto de datos de inicio.

- Desde la perspectiva de obtener reglas descriptivas a partir del modelo generado mediante el conjunto seleccionado, se evalúa la calidad en descubrimiento de subgrupos. En este caso, considerando como medidas el nivel de cobertura, completitud, atipicidad y relevancia, podemos destacar la efectividad del algoritmo evolutivo estratificado, el cual supera notablemente al resto de los métodos en la calidad de sus resultados.

B. Líneas de Investigación Futuras

A continuación, se presentan algunas líneas de trabajo futuras que se plantean a partir de los métodos propuestos en esta memoria.

B.1 Uso de Algoritmos Genéticos Multiobjetivo con Doble Objetivo, Reducción y Precisión, para la Selección de Prototipos

En el Capítulo 2, hemos visto como la función objetivo empleada en los `AAEE` combinaba de forma ponderada la precisión y reducción conseguida en cada solución.

Al combinar dichos factores al 50 % en un mismo valor, se pondera del mismo modo a ambos objetivos. Esta circunstancia, le supone la misma dificultad a reducir el tamaño del conjunto que a incrementar la precisión conseguida, cuando no es así. Resulta mucho más sencillo el reducir el tamaño que el incrementar la precisión, lo cual ocasiona que se tienda a escoger conjuntos de menor tamaño a costa de su precisión. Empíricamente hemos encontrado que siguiendo nuestro modelo de combinación en la función objetivo el equilibrio entre ambas se com-

portaba adecuadamente. Una segunda posibilidad es considerar ambos objetivos de forma independiente mediante la búsqueda de soluciones no dominadas, lo cual podría mejorar los resultados conseguidos.

Por lo tanto, se propone considerar dos objetivos, precisión y simplicidad, mediante un `AAEE` multiobjetivo [CVL02, Deb01]. En cualquier problema con múltiples objetivos, siempre hay un conjunto de soluciones que son superiores a las demás en el espacio de búsqueda cuando se consideran todos los objetivos. Dichas soluciones se conocen como soluciones no dominadas (conjunto Pareto). Ninguna de las soluciones contenidas en el conjunto Pareto es absolutamente mejor que el resto de las no dominadas. De esta manera, se podría obtener un conjunto de soluciones que abarcaría desde los modelos más precisos hasta los más simples, pasando por distintos niveles de equilibrio entre ambos criterios.

B.2 Combinación de la Selección de Instancias con la Selección de Características

La reducción evolutiva combinada de instancias y características ha sido previamente estudiada en la literatura [INN01, LG03]. Habitualmente los estudios desarrollados en este ámbito han ido destinados a aumentar las prestaciones de clasificadores basados en instancias.

Nos planteamos analizar los subconjuntos de instancias y características seleccionados tanto desde la perspectiva de la precisión e interpretabilidad en los modelos predictivos, como desde la novedad, atipicidad, relevancia, etc., de las reglas que componen los modelos descriptivos para el descubrimiento de subgrupos.

Ya hemos podido comprobar que la selección de instancias evaluada de forma independiente presenta un comportamiento destacado. En el estudio futuro se pretenderá analizar si su combinación con selección de características muestra el mismo comportamiento.

B.3 Selección Evolutiva de Prototipos en Problemas con Clases No Balanceadas

El proceso de descubrimiento de conocimiento puede ser aplicado sobre conjuntos de datos que presenten clases minoritarias, donde su presencia o ausencia

en el subconjunto seleccionado no afecten sensiblemente a nivel de precisión de su clasificador asociado. Sin embargo, dichas clases pueden ser suficientemente importantes como para no poder obviar su eliminación en el proceso de selección. Sería por tanto necesario modificar la estrategia de selección para adaptarse a estos casos [BSGR03].

Una posibilidad sería introducir restricciones en las soluciones codificadas en cada cromosoma de tal forma que cada solución obtenida presentara un porcentaje de muestras seleccionadas de cada clase equivalente al porcentaje que aparece en el conjunto de partida. De esta manera podríamos asegurar la presencia de muestras de todas las clases en el subconjunto seleccionado final.

B.4 Análisis de las Prestaciones de los Mecanismos de Poda en Árboles de Decisión Frente a la Selección Evolutiva de Conjuntos de Entrenamiento

Tras llevar a cabo la generación de modelos representativos mediante árboles de decisión, gran parte de los algoritmos de dedicados a este fin efectúan poda del árbol. Con esta poda se pretende mejorar la capacidad de generalizar del modelo, intentando evitar el sobreajuste a partir de los datos de entrenamiento.

Este proceso reduce el tamaño del árbol, disminuyendo el número de hojas y la profundidad (número de antecedentes si lo convertimos en reglas) de cada rama que presenta. La fase de poda puede llevarse a cabo durante la generación del árbol o bien en una etapa posterior de postprocesamiento.

Si los datos de los que se parte para generar el árbol no son de suficiente calidad, el modelo generado puede no serlo tampoco. Frente a dicha circunstancia el mecanismo de poda poco puede hacer para mejorar las prestaciones del modelo.

El estudio futuro pretende analizar las prestaciones que ofrecería un subconjunto de entrenamiento de mayor calidad empleado para generar árboles de decisión frente a los mecanismos de poda. Los objetivos perseguidos serían los siguientes:

- Comprobar si la selección evolutiva de instancias puede ofrecer mejores prestaciones en los modelos generados que los modelos obtenidos aplicando tan solo mecanismos de poda. En este caso como prestaciones podemos considerar la capacidad de predicción del modelo.

- Comparar los modelos obtenidos siguiendo ambas vías, poda y selección evolutiva, con objeto de analizar la interpretabilidad de los modelos resultantes. Tan importante es la precisión del modelo como la posibilidad de extraer de él información fácilmente interpretable y utilizable por parte de usuarios finales.

- Plantear la combinación de selección evolutiva de instancias y mecanismos de poda para aumentar la calidad de los modelos. Dado que ambas vías no son incompatibles, es posible que la combinación de ambas mejoren sus resultados individuales.

B.5 Analisis e Hibridación de Técnicas Nuevas de Selección de Instancias

Los mecanismos de selección de instancias empleados en esta memoria sirvieron de base para el estudio publicado en [CHL03b]. Este estudio se llevo a cabo en la primera fase de desarrollo de esta memoria, con lo que se consideraron en él los algoritmos presentes hasta la fecha.

Otra vía de estudio futura a corto plazo es confeccionar un nuevo análisis con las nuevas técnicas que se han publicado recientemente [RAT03, ZS02, ZZGZ03].

Se estudiará al mismo tiempo las posibles hibridaciones de la selección evolutiva de instancias con ellas, tratando de aprovechar las características específicas de cada técnica para obtener un conjunto de instancias que represente una adecuada distribución de la base de datos.

B.6 Analisis del Empleo de los Algoritmos Evolutivos con Buen Equilibrio entre Diversidad y Convergencia

Los `AAEE` ven afectada su rapidez de convergencia cuando se aplican en la `SII` sobre conjuntos de datos de tamaño elevado.

En este estudio se ha destacado el algoritmo `CHC` de entre los `AAEE` analizados por su excelente comportamiento, siendo una de sus características principales el fomentar la diversidad en la población, lo que puede incrementar su velocidad de convergencia hacia soluciones efectivas.

Se plantea por tanto como estudio futuro el analizar el empleo de `AAEE` que

presenten un buen equilibrio entre diversidad y convergencia con el objetivo de optimizar el proceso de búsqueda [LHC04].

Bibliografía

[AD91] Almuallim H. y Dietterich T. (1991) Learning with many irrelevant features. En *Proceedings of the Ninth National Conference on Artificial Intelligence*, volumen 2, páginas 547–55. AAAI Press.

[Agu01] Aguilar J. S. (2001) *Generación de reglas jerárquicas de decisión con algoritmos evolutivos en aprendizaje supervisado.* PhD thesis, Universidad de Sevilla.

[AKA91] Aha D. W., Kibbler D., y Albert M. K. (1991) Instance based learning algorithms. *Machine Learning* 6: 37–66.

[ALF00] Araujo D. L. A., Lopes H. S., y Freitas A. A. (2000) Rule discovery with a parallel genetic algorithm. En Freitas A. A., Hart W., Krasnogor N., y Smith J. (Eds.) *Data Mining with Evolutionary Algorithms*, páginas 89–94.

[AMS+96] Agrawal R., Mannila H., Srikant R., Toivonen H., y Verkamo A. I. (1996) *Advances in knowledge discovery and data mining*, chapter Fast discovery of association rules, páginas 307–328. American Association for Artificial Intelligence.

[ARBD04] Aguilar-Ruiz J., Bacardit J., y Divina F. (2004) Experimental evaluation of discretization schemes for rule induction. En *Proc. of the Genetic Evolutionary Computation- GECCO 2004, Genetic and Evolutionary Computation Conference*, páginas 828–839.

[Bal94] Baluja S. (1994) Population-based incremental learning. Technical report, Carnegie Mellon University. Technical Report CMU-CS-94-163.

[BEYL04] Baram Y., El-Yaniv R., y Luz K. (2004) Online choice of active learning algorithms. *Journal of Machine Learning Research* (5): 255–291.

[Bez81] Bezdek J. C. (1981) *Pattern recognition with fuzzy objective function algorithms*. Plenum Press.

[BFH01] Baohua G., Feifang H., y Huan L. (2001) Sampling: Knowing whole from its parts. En Liu H. y Motoda H. (Eds.) *Instance Selection and Construction for Data Mining*, páginas 21–38. Kluwer Academic Publishers.

[BFM97] Back T., Fogel D., y Michalewicz Z. (1997) *Handbook of evolutionary computation*. Oxford University Press.

[BG02] Bacardit J. y Garrell J. M. (2002) Métodos de generalización para sistemas clasificadores de pittsburgh. En *Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados*, páginas 486–493.

[BG03] Bacardit J. y Garrell J. M. (2003) Incremental learning for pittsburgh approach classifier systems. En *Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, páginas 303–311.

[BH67] Ball G. H. y Hall D. J. (1967) ISODATA, an iterative method of multivariate analysis and pattern classification. *Behavioral Science* 12: 153–155.

[BL94] Bhanu B. y Lee S. (1994) *Genetic Learning for adaptive image segmentation*. Kluwer Academic Publishers.

[BM02] Brighton H. y Mellish C. (2002) Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery* 6: 153–172.

[Bot00] Bot M. C. J. (2000) Improving induction of linear classification trees with genetic programming. En *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, páginas 403–410. Morgan Kaufmann.

[Bou04] Boulle M. (2004) Khiops: A statistical discretization method of continuous attributes. *Machine Learning* 55(1): 53–69.

[Bra02] Bramer M. (2002) Using J-pruning to reduce overfitting in classification trees. *Knowledge-Based Systems* 15: 301–308.

[Bre96] Breiman L. (1996) Bagging predictors. *Machine Learning* 24(2): 123–140.

[Bro93] Broadley C. E. (1993) Addresing the selective superiority problem: automatic algorithm/model class selection. En *Proceedings of the Tenth International Machine Learning Conference*, páginas 17–24.

[BSGR03] Barandela R., Sánchez J. S., García V., y Rangel E. (2003) Strategies for learning in class imbalance problems. *Pattern Recognition* 36: 849–851.

[BZP94] Bhandarkar S., Zhang Y., y Potter W. (1994) An edge detection technique using genetic algorithms. *Pattern Recognition* (27): 1159–1180.

[CAL94] Cohn D. A., Atlas L., y Ladner R. E. (1994) Improving generalization with active learning. *Machine Learning* 15(2): 201–221.

[Cat91] Catlett J. (1991) On changing continuous attributes into ordered discrete attributes. En *Proceeding of the Fifth European Working Session on Learning*, páginas 164–177. Springer-Verlag.

[CB91] Clark P. y Boswell R. (1991) Rule induction with CN2: Some recent improvements. En *Proc. Fifth European Working Session on Learning*, páginas 151–163. Springer, Berlin.

[CBS91] Chan C. C., Batur C., y Srinivasan A. (1991) Determination of quantization intervals in rule based model for dynamic. En *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, páginas 1719–1723.

[CCdJH01] Casillas J., Cordón O., del Jesus M. J., y Herrera F. (2001) Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. *Information Sciences* 136: 169–191.

[CCK+99] Chapman P., Clinton J., Khabaza T., Reinart T., y Wirth R. (1999) The crisp-dm process model. www.crisp-dm.org.

[CDPY97] Cagnoni S., Dobrzeniecki A., Poli R., y Yanch J. (1997) Segmentation of 3D medical images through genetically-optimized contour-tracking algorithms. Technical report CSRP-97-28, University of Birmingham School of Computer Sicence.

[CGB94] Chmielewski M. R. y Grzymala-Busse J. W. (1994) Global discretization of attributes as preprocessing for machine learning. En *Proceeding of the Third International workshop on Rough Sets and Soft Computing*, páginas 474–480.

[CGJ95] Cohn D. A., Ghahramani Z., y Jordan M. I. (1995) Active learning with statistical models. En Tesauro G., Touretzky D., y Leen T. (Eds.) *Advances in Neural Information Processing Systems*, volumen 7, páginas 705–712. The MIT Press.

[CH67] Cover T. y Hart P. (1967) Nearest neighbour classification. *IEEE Transaction on Information Theory* 13(1): 21–27.

[CHHM01] Cordón O., Herrera F., Hoffmann F., y Magdalena L. (2001) *GENETIC FUZZY SYSTEMS. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Vol. 19 of Advances in Fuzzy Systems - Applications and Theory. World Scientific.

[CHL02] Cano J. R., Herrera F., y Lozano M. (2002) Selección evolutiva de instancias en minería de datos: un estudio experimental. En Herrera F., Riquelme J., y Aguilar J. (Eds.) *Actas del Workshop de Minería de Datos y Aprendizaje*, páginas 137–152.

[CHL03a] Cano J. R., Herrera F., y Lozano M. (2003) Evolutionary algorithms in stratified instance selection for data reduction in large datasets. En *Proc. of the 8th Online World Conference on Soft Computing in Industrial Applications*.

[CHL03b] Cano J. R., Herrera F., y Lozano M. (2003) Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE Transaction on Evolutionary Computation* 7(6): 561–575.

[CHL04a] Cano J. R., Herrera F., y Lozano M. (2004) Estratificación y selección evolutiva de prototipos aplicadas a bases de datos de gran

tamaño. En *Actas del Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, páginas 211–220.

[CHL04b] Cano J. R., Herrera F., y Lozano M. (2004) *Knowledge Discovery in Advances Information Systems*, chapter Instance selection using evolutionary algorithms: an experimental study. Springer Verlag. Aceptado.

[CHL04c] Cano J. R., Herrera F., y Lozano M. (2004) Stratification for scaling up evolutionary prototype selection. *Pattern Recognition Letters.* Aceptado.

[CHS98] Cordón O., Herrera F., y Sánchez L. (1998) *Genetic Algorithms and Evolution Strategies in Engineering and Computer Sciences. Recent Advances and Industrial Applications*, chapter Evolutionary learning processes for data analysis in electrical engineering applications, páginas 205–224.

[CHS99] Cordón O., Herrera F., y Sánchez L. (1999) Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence* 10: 5–24.

[CM97] Cerquides J. y Mantaras R. L. (1997) Proposal and empirical comparison of a parallelizable distance-based discretization method. En *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, páginas 139–142.

[CN89] Clark P. y Niblett T. (1989) The CN2 induction algorithm. *Machine Learning* 3(4): 261–283.

[Coh95] Cohen W. W. (1995) Fast effective rule induction. En *Proceedings of the Fifth European Working Session on Learning*, páginas 151–163. Springer.

[COMV04] Castejón M., Ordieres J. B., Martínez F. J., y Vergara E. P. (2004) Outlier detection and data cleaning in multivariate non-normal samples: The PAELLA algorithm. *Data Mining and Knowledge Discovery* (9): 171–187.

[CVL02] Coello C. A., Veldhuizen D. A. V., y Lamont G. B. (2002) *Evolutionary algorithms for solving multi-objective problems.* Kluwer Academic Publishers.

[Das97] Dash M. (1997) Feature selecion via set cover. En *Proceedings of IEEE Knowledge and Data Engineering Exchange Workshop*, páginas 165–171. IEEE Computer Society.

[DDBM03] Detours V., Dumont J. E., Bersini H., y Maenhaut C. (2003) Integration and cross-validation of high-throughtput gene expression data: comparing heterogeneous data sets. *FEBS Letters* 546(1): 98–102.

[Deb01] Deb K. (2001) *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

[DK82] Devijver P. A. y Kittler J. (1982) *Pattern Recognition: A Statistical Approach*. Prentice-Hall.

[DKS95] Dougherty J., Kohavi R., y Sahami M. (1995) Supervised and unsupervised discretization of continuous features. En *Proceedings of the Twelfth International Conference on Machine Learning*, páginas 194–202.

[DL98] Dash M. y Liu H. (1998) Hybrid search of feature subsets. En *Pacific Rim International Conference on Artificial Intelligence*, páginas 238–249.

[dM93] de Merckt T. V. (1993) Decision trees in numerical attribute spaces. En *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, páginas 1016–1021.

[Doa92] Doak J. (1992) An evaluation of feature selection methods and their application to computer security. Technical report, University of California at Davis. Technical Report CSE-92-18.

[DP96] Domingos P. y Pazzani M. (1996) Beyond independence: Conditions for the optimality of the simple bayesian classifier. En *Proceedings of the Thirteenth International Conference on Machine Learning*, páginas 103–130. Morgan Kaufmann.

[DSG93] DeJong K. A., Spears W. M., y Gordon D. F. (1993) Using genetic algorithms for concept learning. *Machine Learning* 2-3(13): 161–188.

[DuM01] DuMouchel W. (2001) *Handbook of Massive Data Sets*, chapter Data squashing: constructing summary data sets, páginas 579–591. Kluwer Academic Publishers.

[Dv04] Džeroski S. y Ženko B. (2004) Is combining classifiers with stacking better than selecting the best one? *Machine Learning* 54(3): 255–273.

[DVJ+99] DuMouchel W., Volinsky C., Johnson T., Cortes C., y Pregibon D. (1999) Squashing flat files flatter. En *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining*, páginas 6–15.

[EPE04] Evgeniou T., Pontil M., y Elisseeff A. (2004) Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning* 55: 71–97.

[Esh91] Eshelman L. J. (1991) The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of Genetic Algorithms* 1: 265–283.

[FAV99] Ferri F. J., Albert J. V., y Vidal E. (1999) Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules. *IEEE Transactions on Systems, Man, and Cybernetics* 29 (Part B)(4): 667–672.

[FI93a] Fayyad U. y Irani K. (1993) Multi-interval discretization of continuous attributes as preprocessing for classification learning. En *Proc. of the 13th International Join Conference on Artificial Intelligence*, number 1022–1027.

[FI93b] Fayyad U. y Irani K. (1993) Multi-interval discretization of continuous-valued attributes for classification learning. En *Proceeding of the Thirteenth International Joint Conference on Artificial Intelligence*, páginas 1022–1027. Morgan Kaufmann.

[FPSS96] Fayyad U. M., Piatetsky-Shapiro G., y Smyth P. (1996) From data mining to knowledge discovery: an overview. En Fayyad U. M., Piatetsky-Shapiro G., Smyth P., y Uthurusamy R. (Eds.) *Advances in Knowledge Discovery and Data Mining*, páginas 495–515. The MIT Press.

[Fre95] Freund Y. (1995) Boosting a weak learning algorithm by majority. *Information and Computation* 121(2): 256–285.

[Fre02] Freitas A. A. (2002) *Data mining and knowledge discovery with evolutionary algorithms.* Springer-Verlag.

[GARR+02] Giráldez R., Aguilar-Ruiz J., Riquelme J., Ferrer-Troyano F., y Rodríguez D. (2002) Discretization oriented to decision rules generation. *Frontiers in Artificial Intelligence and Applications* 82: 275–279.

[Gat72] Gates G. W. (1972) The reduced nearest neighbour rule. *IEEE Transaction on Information Theory* 18(5): 431–433.

[GL02] Gamberger D. y Lavrač N. (2002) Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research* 17: 501–527.

[Gol89] Goldberg D. E. (1989) *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley.

[Gol02] Goldberg D. E. (2002) *The design of competent genetic algorithms: Steps toward a computational theory of innovation.* Kluwer Academic Pub.

[Gra04] Grandvalet Y. (2004) Bagging equalizes influence. *Machine Learning* 55: 251–270.

[GS93] Giordana A. y Saitta L. (1993) Regal: an integrated system for learning relations using genetic algorithms. En *Proceedings of the Second International Workshop on Multistrategy Learning*, páginas 234–249.

[GSW99] Guerra-Salcedo C. y Whitley D. (1999) Genetic approach to feature selection for ensemble creation. En *Proceedings of the Genetic and Evolutionary Computation Conference*, páginas 236–243.

[Har68] Hart P. E. (1968) The condensed nearest neighbour rule. *IEEE Transaction on Information Theory* 18(3): 431–433.

[HCBB02] Hall L. O., Collins R., Bowyer K. W., y Banfield R. (2002) Error-based pruning of decision trees grown on very large data sets can

work! En *International Conference on Tools for Artificial Intelligence*, páginas 233–238.

[HLL02] Ho S.-Y., Liu C.-C., y Liu S. (2002) Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. *Pattern Recognition Letter* (23): 1495–1503.

[HMS01] Hand D. J., Mannila H., y Smyth P. (2001) *Principles of data mining (adaptive computation and machine learning)*. MIT Press.

[Ho89] Ho T. (1989) The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (8): 832–844.

[Hol75] Holland J. H. (1975) *Adaptation in natural and artificial systems*. The University of Michigan Press.

[Hol93] Holte R. C. (1993) Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11: 63–90.

[HR02] Hasenjager M. y Ritter H. (2002) *New learning paradigms in soft computing*, chapter Active learning in neural networks, páginas 137–169. Physica-Verlag GmbH.

[HRF04] Hernández J., Ramirez M. J., y Ferri C. (2004) *Introducción a la Minería de Datos*. Pearson.

[HS97] Ho K. M. y Scott P. D. (1997) Zeta: A global method for discretization of continuous variables. En *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, páginas 191–194.

[ILS01] Inza I., Larrañaga P., y Sierra B. (2001) Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms. *International Journal of Approximate Reasoning* 27(2): 143–164.

[IML+01] Inza I., Merino M., Larrañaga P., Quiroga J., Sierra B., y Girala M. (2001) Feature subset selection by genetic algorithms and estimation of distribution algorithms: A case study in the survival of cirrhotic patients treated with TIPS. *Artificial Intelligence in Medicine* 23(2): 187–205.

[INN01] Ishibuchi H., Nakashima T., y Nii M. (2001) *Instance Selection and Construction for Data Mining*, chapter Genetic-algorithm-based instance and feature selection, páginas 95–113. Kluwer Academic Publishers.

[Jan93] Janikow C. Z. (1993) A knowledge-intensive genetic algorithm for supervised learning. *Machine Learning* 1(13): 169–228.

[JD88] Jain A. K. y Dubes R. C. (1988) *Algorithms for Clustering Data*. Prentice Hall.

[JMF99] Jain A. K., Murty M.Ñ., y Flynn P. J. (1999) Data clustering: a review. *ACM Computing Surveys* 31(3): 264–323.

[KA87] Kibbler D. y Aha D. W. (1987) Learning representative exemplars of concepts: An initial case of study. En *Proc. of the Fourth International Workshop on Machine Learning*, páginas 24–30.

[KCH+03] Kim W., Choi B., Hong E., Kim S., y Lee D. (2003) A taxonomy of dirty data. *Data Mining and Knowledge Discovery* 7: 81–99.

[Ker92] Kerber R. (1992) Chimerge: discretization of numeric atributes. En *Proceeding of the Ninth National Conference Artificial Intelligence*, páginas 123–128. The MIT Press.

[Kin67] King B. (1967) Step-wise clustering procedures. *J. Am. Stat. Assoc.* (69): 86–101.

[KJ99] Kuncheva L. I. y Jain L. C. (1999) Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters* (20): 1149–1156.

[Klö96] Klöesgen W. (1996) *Advances in knowledge discovery and data mining*, chapter Explora: A multipattern and multiestrategy discovery assistant, páginas 249–271. MIT Press.

[KME04] Kääriäinen M., Malinen T., y Elomaa T. (2004) Selective rademacher penalization and reduced error pruning of decision trees. *Journal of Machine Learning Research* 5: 1107–1126.

[KMOB04] Kweku-Muata y Osei-Bryson (2004) Evaluation of decision trees: a multicriteria aproach. *Cumputers and Operations Research* 31: 1933–1945.

[KN90] Kabada N. y Nygard K. E. (1990) Improving the perfomance of genetic algorithms in automated discovery of parameters. En *Machine Learning: Proceedings of the Seventh International Conference*, páginas 140–148.

[Koz92] Koza J. R. (1992) *Genetic Programming: on the programming of computers by means of natural selection.* The MIT Press.

[KR88] Krishnaiah P. y Rao C. (1988) *Handbook of statistics 6: sampling.* North-Holland.

[KR92] Kira K. y Rendell L. (1992) A practical approach to feature selection. En *Proceedings of the Ninth International Conference on Machine Learning*, páginas 249–256.

[KS03] Klivans A. R. y Servedio R. A. (2003) Boosting and hard-core set construction. *Machine Learning* 51: 217–238.

[Kun95] Kuncheva L. (1995) Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters* 16: 809–814.

[LCGF04] Lavrač N., Cestnik B., Gamberger D., y Flach P. (2004) Decision support through subgroup discovery: Three case studies and the lessons learned. *Machine Learning* 57: 115–143.

[LF78] Lu S. Y. y Fu K. S. (1978) A sentence-to-sentence clustering procedure for pattern analisys. *IEEE Trans. Syst. on Man Cybernetics* (8): 381–389.

[LFZ99] Lavrač N., Flach P., y Zupan B. (1999) Rule evaluation measures: A unifying view. En *Proceeding of the Nineth International Workshop on Inductive Logic Programming*, páginas 74–185. Springer.

[LG99] Llorá X. y Garrell J. M. (1999) GENIFER: A nearest neighbor based classifier system using GA. En *Proced. of the Genetic and Evolutionary Conference*, página 797. Morgan Kaufmann Publishers.

[LG03] Llorá X. y Garrell J. M. (2003) Prototype induction and attribute selection via evolutionary algorithms. *Intelligent Data Analysis* 7(3): 193–208.

[LHC04] Lozano M., Herrera F., y Cano J. R. (2004) Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Information Sciences* Aceptado.

[LHTD02] Liu H., Hussain F., Tan C. L., y Dash M. (2002) Discretization: an enabling technique. *Data Mining and Knowledge Discovery* 6: 393–423.

[Lin02] Lin T. Y. (2002) Attribute transformation for data mining I: Theoretical explorations. *International Journal of Intelligent Systems* 17: 213–222.

[LKFT04] Lavrač N., Kavšec B., Flach P., y Todorovski L. (2004) Subgroup discovery with CN2-SD. *Journal of Machine Learning Research* (5): 153–188.

[LL96] Larrañaga P. y Lozano J. A. (1996) Structure learning of bayesian networks by genetic algorithms: A perfomance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Inteligence* 18(9): 912–926.

[LL99] Lozano J. A. y Larrañaga P. (1999) Applying genetic algorithms to search for the best hierarchical clustering of a data set. *Pattern Recognition Letters* 20(9): 911–918.

[LL01] Larrañaga P. y Lozano J. A. (2001) *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation.* Kluwer Academic Publishers.

[LM98a] Liu H. y Motoda H. (Eds.) (1998) *Feature Extraction, Construction and Selection: A Data Mining Perspective.* Kluwer Academic Publishers.

[LM98b] Liu H. y Motoda H. (1998) *Feature selection for knowledge discovery and data mining.* Kluwer Academic Publishers.

[LM01a] Liu H. y Motoda H. (2001) Data reduction via instance selection. En Liu H. y Motoda H. (Eds.) *Instance Selection and Construction for Data Mining*, páginas 3–20. Kluwer Academic Publishers.

[LM01b] Liu H. y Motoda H. (Eds.) (2001) *Instance selection and construction for data mining.* Kluwer Academic Publishers.

[LM02] Liu H. y Motoda H. (2002) On issues of instance selection. *Data Mining and Knowledge Discovery* 6: 115–130.

[LMD98] Liu H., Motoda H., y Dash M. (1998) A monotonic measure for optimal feature selection. En *Proceedings of the European Conference on Machine Learning*, páginas 101–106.

[Low95] Lowe D. G. (1995) Similarity metric learning for a variable-kernel classifier. *Neural Computation* 7(1): 72–85.

[LS95] Liu H. y Setiono R. (1995) Chi2: feature selection and discretization of numeric attributes. En *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence*, páginas 388–391.

[LS96a] Liu H. y Setiono R. (1996) Feature selection and classification - a probabilistic wrapper approach. En *Proceedings of the Ninth International Conference on Industrial and Ingineering Applications of AI and ES*, páginas 419–424.

[LS96b] Liu H. y Setiono R. (1996) A probabilistic approach to feature selection - a filter solution. En *Proceedings of International Conference on Machine Learning*, páginas 319–327. Morgan Kaufmann Publishers.

[LS98] Liu H. y Setiono R. (1998) Incremental feature selection. *Applied Intelligence* 9(3): 217–230.

[LSG+97] Larrañaga P., Sierra B., Gallego M. Y., Michelena M. J., y Pikaza J. M. (1997) Learning bayesian networks by genetic algorithms. a case study in the prediction of survival in malignant skin melanoma. *Lecture Notes in Artificial Intelligence* 1211: 261–272.

[Man02] Mandischer M. (2002) A comparison of evolution strategies and backpropagation for neural network training. *Neurocomputing* 42: 87–117.

[MBL04] Miquélez T., Bengoetxea E., y Larrañaga P. (2004) Evolutionary computation based on bayesian classifiers. *International Journal of Applied Mathematics and Computer Science* 14(3): 101–115.

[McQ67] McQueen J. (1967) Some methods for classification and analisys of multivariate observations. En *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistic and Probability*, páginas 281–287.

[MM96] Merz C. J. y Murphy P. M. (1996) UCI repository of machine learning databasesUniversity of California Irvine, Department of Information and Computer Science, http://kdd.ics.uci.edu.

[MRD+02] Madigan D., Raghavan N., Domouchel W., Nason M., Posse C., y Ridgeway G. (2002) Likelihood-based data squashing: a modeling approach to instance construction. *Data Mining and Knowledge Discovery* 6: 173–190.

[Mur84] Murtagh F. (1984) A survey of recent advances in hierarchical clustering algorithms which use cluster centers. *Comput. J.* (26): 354–359.

[NF77] Narendra P. y Fukunaga K. (1977) A branch and bound algorithm for feature subset selection. *IEEE Transaction on Computers* C-26(9): 917–922.

[NI98] Nakashima T. y Ishibuchi H. (1998) GA-based approaches for finding the minimum reference set for nearest neighbour classification. En *Proceedings of the IEEE International Conference on Evolutionary Computation*, páginas 709–714.

[NS98] Nikolaev N. I. y Slavov V. (1998) Inductive genetic programation with decision trees. *Intelligent Data Analysis* (1): 31–44.

[NS01] Nock R. y Sebban M. (2001) Advances in adaptive prototype weighting and selection. *International Journal on Articificial Intelligence Tools* 10(1-2): 137–155.

[Owe03] Owen A. B. (2003) Data squashing by empirical likelihood. *Data Mining and Knowledge Discovery* 7(1): 101–113.

[PGP+93] Punch W., Goodman E., Pei M., Lai C., Hovland P., y Enbody R. (1993) Further research on feature selection and classification using genetic algorithms. En *Proceedings of the Fifth International Conference on Genetic Algorithms*, páginas 557–564.

[PJO99] Provost F. J., Jensen D., y Oates T. (1999) Efficient progressive sampling. En *Proceedings of the Fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, páginas 23–32.

[Pyl99] Pyle D. (1999) *Data preparation for data mining.* Morgan Kaufmann.

[Qui86] Quinlan J. R. (1986) Induction of decision trees. *Machine Learning* 1(1): 81–106.

[Qui93] Quinlan J. R. (1993) *C4.5: Programs for machine learning.* Morgan Kaufmann.

[Qui96] Quinlan J. R. (1996) Bagging, boosting, and C4.5. En *Proceedings of the AAAI/IAAI*, volumen 1, páginas 725–730.

[RAT00] Riquelme J. C., Aguilar J., y Toro M. (2000) Discovering hierarchical decision rules with evolutionary algorithms. *International Journal of Computer, Systems and Signals* 1(1): 73–84.

[RAT03] Riquelme J. C., Aguilar J. S., y Toro M. (2003) Finding representative patterns with ordered projections. *Pattern Recognition* 36(4): 1009–1018.

[RB01] Reeves C. R. y Bush D. R. (2001) *Instance Selection and Construction for Data Mining*, chapter Using genetic algorithms for training data selection in RBF networks, páginas 339–356. Kluwer Academic Publishers.

[RN01] Ravindra T. y Narasimha M. (2001) Comparison of genetic algorithm based prototype selection schemes. *Pattern Recognition* 34: 523–525.

[ROM01] Ratsch G., Onoda T., y Muller K.-R. (2001) Soft margins for adaboost. *Machine Learning* (42): 287–320.

[RRA02] Ruiz R., Riquelme J. C., y Aguilar J. S. (2002) Projection-based measure for efficient feature selection. *Journal of Intelligent and Fuzzy Systems* 12(3-4): 175–183.

[Rus69] Ruspini E. H. (1969) A new approach to clustering. *Inf. Control* (15): 22–32.

[RWLI75] Ritter G. L., Woodruff H. B., Lowry S. R., y Isenhour T. L. (1975) An algorithm for a selective nearest neighbour decision rule. *IEEE Transaction on Information Theory* 21(6): 665–669.

[SB95] Skinner A. y Broughton J. Q. (1995) Neural networks in computational material science: training algorithms. *Modeling and Simulation in Material Science and Engineering* páginas 371–390.

[SB02] Sarawagi S. y Bhamidipaty A. (2002) Interactive deduplication using active learning. En *Proceedings of the The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[SC01] Sánchez L. y Corrales J. A. (2001) *Mathware and Soft Computing*, volumen VII, chapter A niching scheme for steady state GA-P and its application to fuzzy rule based classifiers induction, páginas 337–350.

[SCC01] Sánchez L., Couso I., y Corrales J. A. (2001) Comparing GP operators with SA search to evolve fuzzy rule classifiers. *Information Sciences* (1–4): 175–192.

[Sch99] Schapire R. E. (1999) Theoretical views of boosting and applications. En *Algorithmic Learning Theory, 10th International Conference, ALT '99, Tokyo, Japan, December 1999, Proceedings*, volumen 1720, páginas 13–25. Springer.

[Ska94] Skalak D. B. (1994) Prototype and feature selection by sampling and random mutation hill climbing algorithms. En *International Conference on Machine Learning*, páginas 293–301.

[SMO96] Scheaffer R., Mendenhall W., y Ott L. (1996) *Elementary survey sampling*. Duxbury Press.

[Sán00] Sánchez L. (2000) Interval-valued GA-P algorithms. *IEEE Transactions on Evolutionary Computation* 4(1): 64–72.

[SS73] Sneath P. H. A. y Sokal R. R. (1973) *Numerical Taxonomy*. Freeman.

[SS88] Siedlecki W. y Sklansky J. (1988) On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence* 2(2): 197–220.

[SS89] Siedlecki W. y Sklansky J. (1989) A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* (10): 335–347.

[SSS04] Schallehn E., Sattler K., y Saake G. (2004) Efficient similarity-based operations for data integration. *Data and Knowledge Engineering* (48): 361–387.

[ST04] Saar-Tsechansky M. (2004) Active sampling for class probability estimation and ranking. *Machine Learning* 54: 153–178.

[SYCCS02] Shinn-Ying H., Chia-Cheng L., y Soundy L. (2002) Design of an optimal nearest neighbour classifier using an intelligent genetic algorithm. *Pattern Recognition Letters* 23(13): 1495–1503.

[Sym77] Symon M. J. (1977) Clustering criterion and multi-variate normal mixture. *Biometrics* (77): 35–43.

[TG96] Tumer K. y Ghosh J. (1996) Classifier combining: analytical results and implications. En *Proceedings of the AAAI 96 - Workshop in Induction of Multiple Learning Models.*

[VSMdB97] Venturini G., Slimane M., Morin F., y de Beauville J.-P. A. (1997) On using interactive genetic algorithms for knowledge discovery in databases. En *Proceedings of the Seventh International Conference on Genetic Algorithms*, páginas 696–703. Morgan Kaufmann.

[Wee96] Weeks A. (1996) *Fundamentals of electronic image processing.* The International Society for Optical Engineering Press.

[WF00] Witten I. H. y Frank E. (2000) *Data mining: practical machine learning tools and techniques with java implementations.* Morgan Kaufmann.

[WH89] Whitley D. y Hanson T. (1989) Optimizing neural networks using faster, more accurate genetic search. En *Proceedings of the Third International Conference on Genetic Algorithms*, páginas 391–397.

[Whi89] Whitley D. (1989) The GENITOR algorithm and selective preasure: Why rank based allocation of reproductive trials is best. En *Proceedings of the Third International Conference on GAs*, páginas 116–121. Morgan Kauffman.

[Wil72a] Wilson D. L. (1972) Asymptotic properties of nearest neighbour rules using edited data. *IEEE Transaction on Systems, Man. and Cybernetics* 2: 408–420.

[Wil72b] Wilson D. L. (1972) Asymptotic properties of nearest neighbour rules using edited data. *IEEE Transactions on Systems Man. and Cybernetics* 2: 408–421.

[WL98] Wang K. y Liu B. (1998) Concurrent discretization of multiple attributes. En *Pacific-Rim International Conference on AI*, páginas 250–259.

[WM97] Wilson D. R. y Martinez T. R. (1997) Instance pruning techniques. En *Proceedings of the 14th International Conference on Machine Learning*, páginas 403–411.

[WM00a] Wilson D. R. y Martinez T. R. (2000) An integrated instance-based learning algorithm. *Computational Intelligence* 16(1): 1–28.

[WM00b] Wilson D. R. y Martinez T. R. (2000) Reduction tecniques for instance-based learning algorithms. *Machine Learning* 38: 257–268.

[Wol92] Wolpert D. H. (1992) Stacked generalization. *Neural Networks* 5: 214–259.

[Wro01] Wrobel S. (2001) *Relational Data Mining*, chapter Inductive logic programming for knowledge discovery in databases, páginas 74–101. Springer.

[XYC88] Xu L., Yan P., y Chang T. (1988) Best first strategy for feature selection. En *Proceedings of the Ninth International Conference on Pattern Recognition*, páginas 706–708.

[Yao99] Yao X. (1999) Evolving artificial neural networks. En *Proceedings of the IEEE*, volumen 87, páginas 1423–1447.

[YH98] Yang J. y Honavar V. (1998) Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems* 13: 44–49.

[Zad65] Zadeh L. A. (1965) Fuzzy sets. *Inf. Control* (8): 338–353.

[Zha71] Zhan C. T. (1971) Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Computation* C-20: 68–86.

[Zha00] Zhang B.-T. (2000) Bayesian evolutionary algorithms for learning and optimization. En *Optimization By Building and Using Probabilistic*, páginas 220–223.

[ZJ03] Zhou Z.-H. y Jiang Y. (2003) Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. *IEEE Transactions on Information Technology in Biomedicine* 7(1): 37–42.

[ZS02] Zhang H. y Sun G. (2002) Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recognition* (35): 1481–1490.

[ZZGZ03] Zhao K.-P., Zhou S.-G., Guan J.-H., y Zhou A.-Y. (2003) C-PRUNER: An improved instance pruning algorithm. En *Proc. of the Second International Conference on Machine Learning and Cybernetics*, páginas 94–99.

[ZZY03] Zhang S., Zhang C., y Yang Q. (2003) Data preparation for data mining. *Applied Artificial Intelligence* 17: 375–381.