Técnicas de reducción de datos en KDD. El uso de Algoritmos Evolutivos para la Selección de Instancias

Francisco Herrera
Dpto. de Ciencias de la Comp. e Inteligencia Artificial,
ETS Ingeniería Informática,
Universidad de Granada,
18071 Granada,
herrera@decsai.ugr.es

J.-R. Cano
Dpto. de Informática,
EUP de Linares,
Universidad de Jaén,
23700 Linares (Jaén),
jrcano@ujaen.es

Abstract

En el descubrimiento de información en bases de datos, debido al tamaño de las bases de datos, presencia de ruido, datos inconsistentes, redundantes, etc., se hace necesaria la aplicación de técnicas de preprocesamiento sobre los conjuntos de datos. Dicho preprocesamiento persigue obtener conjuntos de datos tales que al aplicar técnicas de minería de datos sobre ellos se generen modelos representativos con mayores prestaciones. De entre las diferentes tareas que se pueden desarrollar en la etapa de preparación de los datos, centraremos la atención en la reducción de datos y se presentaran las diferentes vías que se pueden seguir para aplicarla. Finalmente, mostramos resultados sobre el uso de los algoritmos evolutivos para la selección de instancias con el objetivo de reducir el conjunto de entrenamiento y extraer árboles de decisión más compactos e interpretables.

Palabras Clave: Selección de Instancias, Algoritmos Evolutivos, Reducción de Datos, Minería de Datos.

1. Introducción

En la actualidad la sociedad se enfrenta al reto de trabajar con volúmenes de información cada vez mayores. El *Descubrimiento de Conocimiento en Bases de Datos* (en inglés Knowledge Discovery in Databases, con el acrónimo KDD que será empleado por ser estándar su uso) es un área de la computación que intenta explotar la ingente cantidad de información, extrayendo conocimiento que pueda asistir a un humano para llevar a cabo tareas de forma más eficiente y satisfactoria.

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 [74]. El objetivo perseguido por el preprocesamiento es obtener conjuntos de datos tales que al aplicar técnicas de minería de datos (MDD [39]) sobre ellos se generen modelos representativos con mayores prestaciones.

Se pueden aplicar diferentes técnicas de reducción de datos, siendo los algoritmos evolutivos una de ellas, ofrenciendo prometedores resultados [77, 78, 89, 12]. En la Sección 9 se presenta un ejemplo de empleo de algoritmos evolutivos para la extración de modelos predictivos más compactos e interpretables.

Este capítulo se organiza en las siguientes secciones. En la Sección 2 se presenta el proceso de preparación de los datos. Una posible vía para llevar cabo dicha preparación consiste en la reducción de datos (RDD), apareciendo descrita en la Sección 3. En las siguientes secciones se estudian las diferentes estrategias de RDD a seguir: selección de características en la Sección 4, selección de instancias en la Sección 5, discretización de características en la Sección 6, agrupamiento de datos en la Sección 7, y compactación de datos en la Sección 8. En la Sección 9 se muestra un ejemplo de preparación de datos mediante reducción para la extracción de árboles de decisión utilizando algoritmos evolutivos. Finalmente, en la Sección 10 se muestran las conclusiones.

2. Preprocesamiento

La preparación o preprocesamiento 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 los procesos de MDD pueda conducir a obtener modelos, patrones o reglas de mayor calidad [74]. La importancia de la preparación de los datos se ve reflejada en los tres aspectos siguientes [101]:

- 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 cual 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.

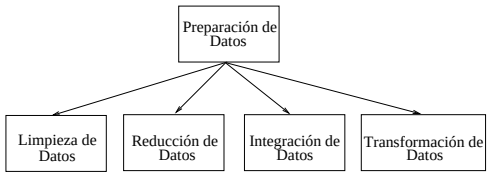

Figura 1: Estrategias para el preprocesado de datos

- 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 [48, 15].
- Reducción de datos (RDD): Consiste en decidir qué 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. Nos centraremos en este caso en esta perspectiva del preprocesamiento: reducir el volumen de datos seleccionando los más relevantes para su posterior uso por algoritmos de MDD [57].

- Integración de datos: Se basa en combinar múltiples tablas o registros para crear nuevos registros o valores. El combinar tablas hace referencia 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 [25, 86].

- 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 [53].

Las estrategias anteriormente descritas no son mutuamente 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).

A continuación centraremos la atención en las técnicas de preprocesado basadas en la RDD.

3. Reducción de Datos

Las técnicas de MDD que extraen modelos a partir de ejemplos tienden a obtener modelos complejos conforme crece el volumen de datos del conjunto sobre el cual se aplican. El elevado tamaño de los conjuntos de datos provoca inconvenientes adicionales tales como:

- Se aumenta el tiempo de respuesta de los modelos. Cuantos más ejemplos se almacenen, mayor será el tiempo necesario para clasificar casos no vistos.

- Se aumenta la sensibilidad al ruido y la posibilidad de sobreajuste de los modelos sobre el conjunto de entrenamiento. Al emplear un mayor número de datos, es más probable que se retengan ejemplos ruidosos. Eso provocará que esos ejemplos de escasa calidad afecten a los modelos modificando la adecuada clasificación de aquellos casos que caigan dentro de su región de decisión.

- Al extraerse modelos de gran tamaño, 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 o reglas para representarla. Cuanto menor sea su tamaño, 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 Figura 2 muestra las diferentes técnicas que se pueden emplear para llevar a cabo la RDD.

En las siguientes secciones procedemos a la descripción de cada una de ellas.

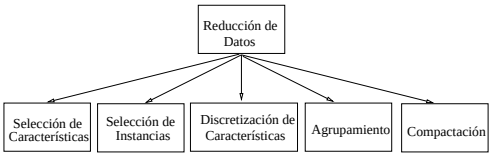

Figura 2: Técnicas de reducción de datos

4. Selección de Características

Existen algoritmos de aprendizaje automático que 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 discriminación, 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.

Los 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.

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 [55, 56]. La Figura 3 refleja el proceso clásico de selección de características.

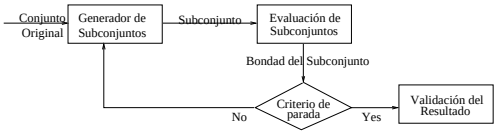

Figura 3: Proceso de selección de características

A continuación vamos a presentar diferentes técnicas empleadas para efectuar la selección de características. Para ello, previamente introducimos diferentes aproximaciones a la clasificación de las mismas.

Una posible forma de clasificar estas técnicas es basarnos en el mecanismo de selección empleado. Tenemos dos aproximaciones: filtro y envoltura. 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.

Un clasificación mas exhaustiva se puede llevar a cabo 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 la 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, de tal forma que inconsistencia cero significa consistencia total.
- 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 (Estocástico): basada en algoritmos de búsqueda globales. Se pretende con ellos no perderse en mínimos locales y encontrar interdependencias entre características que la búsqueda heurística es incapaz de detectar.
- La dirección de búsqueda: es el 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 esquemas previos.

A continuación clasificaremos 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 totalmente el espacio de búsqueda. La Tabla 1 presenta diferentes algoritmos siguiendo esta estrategia.
- Métodos Heurísticos: Son técnicas caracterizadas por sacrificar la promesa del subconjunto solución óptimo a fin de obtener una solución rápida. Para ello emplean el conocimiento disponible para dirigir la búsqueda. En la Tabla 2 se muestran algunos de estos métodos.

Cuadro 1: Métodos de Completitud

Algoritmo	Estrategia	Dirección	Medida	Ref.
Focus	Completa	Adelante	Consistencia	[2]
Aut. Branch & Bound	Completa	Atrás	Consistencia	[59]
Best First	Completa	Adelante	Clásica	[98]
Beam Search	Completa	Adelante	Acierto	[27]
Branch & Bound	Completa	Atrás	Clásica	[71]

Cuadro 2: Métodos Heurísticos

Algoritmo	Estrategia	Dirección	Medida	Ref.
Wrap1	Heurístico	Atrás	Acierto	[56]
SetCover	Heurístico	Adelante	Consistencia	[22]
SOAP	Heurístico	Adelante	Dependencia	[82]

- Métodos Estocásticos: Este tipo de técnicas permiten la búsqueda del subconjunto de características óptimo mediante la generación aleatoria de subconjuntos. En la Tabla 3 aparecen reflejados algunos de estos métodos.

- 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. La Tabla 4 muestra diferentes métodos que siguen esta orientación.

- Métodos Híbridos: Con la hibridación de técnicas se pretende explotar las ventajas de unos métodos, eliminando sus inconvenientes. La Tabla 5 presenta diferentes algoritmos siguiendo esta vía.

- 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. En la Tabla 6 se muestra la técnica representativa.

5. Selección de Instancias

La selección de instancias (`SII`) consiste en escoger las muestras más representativas de un conjunto determinado [47, 9, 58].

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, ya que podemos estar eliminando ejemplos repetidos o ruido. Es interesante el hecho de que cada ejemplo 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 4.

Pasaremos a describir cada una de ellas a continuación:

Cuadro 3: Métodos Estocásticos

Algoritmo	Estrategia	Direc.	Medida	Ref.
Algor. Genét.	No Determ.	Aleat.	Cualquiera	[43]
EDA	No Determ.	Aleat.	Cualquiera	[42]
Enfr. Simulado	No Determ.	Aleat.	Cualquiera	[90]
Las Vegas Filter	No Determ.	Aleat.	Consistencia	[62]
Las Vegas Wrapper	No Determ.	Aleat.	Acierto	[61]

Cuadro 4: Métodos Ponderando Características

Algoritmo	Estrategia	Dirección	Medida	Ref.
Relief	Heurístico	Aleatorio	Clásico	[50]

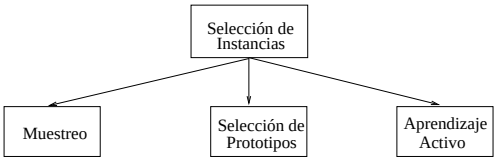

Figura 4: Estrategias de selección de instancias

5.1. 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 [91, 4]: 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 [4]: 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 [4]: 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 [51, 4]: Se

Cuadro 5: Métodos Híbridos

Algoritmo	Estrat.	Direc.	Medida	Ref.
Quick Branch&Bound	No Det.	Aleat.	Consist.	[23]

Cuadro 6: Aproximación Incremental

Algoritmo	Estrategia	Dirección	Medida	Ref.
Las Vegas Incremental	No Determ.	Aleatorio	Consistencia	[63]

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 [87, 4]: 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 [4]: 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 [4]: 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 [73]: 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.

5.2. Selección de Prototipos

Los métodos de selección de prototipos (`SPP`) son técnicas de `SII` que pretenden encontrar conjuntos de instancias 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 `l` 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 que se describen a continuación:

- 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. La Tabla 7 presenta los diferentes algoritmos que

  siguen esta estrategia.

Cuadro 7: Selección basada en el vecino más cercano

Nombre del Algoritmo	Ref.
Condensed Nearest Neighbour (CNN)	[37]
Edited Nearest Neighbour (ENN)	[95]
Repeated Edited Nearest Neighbour (RENN)	[95]
Reduced Nearest Neighbour (RNN)	[34]
Variable Similarity Metric (VSM)	[64]
Multiedit	[26]
Model Class Selection (MCS)	[10]
Shrink	[47]
Instance Based 2 (IB2)	[47]
Instance Based 3 (IB3)	[1]
Iterative Case Filtering (ICF)	[9]
Selective Nearest Neighbour (SNN)	[80]
Gabriel Graph (GG)	[7]
Relative Neighbourhood Graph (RNG)	[7]
PSRCG	[88]

- Selección basada en la métrica Encoding Length. En la Tabla 8 se presentan algunos algoritmos que emplean

  esta métrica.

- Selección basada en eliminación ordenada: En este apartado se encuentran los algoritmos de la familia DROP. Se basan en la eliminación ordenada de instancias y los tenemos reflejados en la Tabla 9.

Cuadro 8: Selección basada en la métrica Encoding Length

Nombre del Algoritmo	Ref.
Elh	[11]
ELGrow	[11]
Explore	[11]
DEL	[97]

- Selección basada en Algoritmos Evolutivos: En este caso la selección se lleva a cabo empleando mecanismos basados en la evolución natural. En la Tabla 10 se muestran algoritmos evolutivos aplicados a SII.

Cuadro 10: Selección basada en Algoritmos Evolutivos

Nombre del Algoritmo	Ref.
Steady-State Genetic Algorithm (SGA)	[52]
Generational Genetic Algorithm (GGA)	[52]
CHC	[12]
PBIL	[12]

- 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 SPP [91]. En la Tabla 11 se muestran algunos de estos métodos.

5.3. Aprendizaje Activo

El proceso de Aprendizaje Activo (*Active Learning* [20, 21, 85, 38, 5, 84]) presenta como diferencia destacable frente a las anteriores técnicas citadas el hecho de que el subconjunto final seleccionado es dinámico.

En aprendizaje activo se parte de un conjunto limitado de muestras ya clasificadas, y otro conjunto mucho mayor sin clasificar. El conjunto clasificado se

Cuadro 9: Selección basada en eliminación ordenada

Nombre del Algoritmo	Ref.
Decremental Reduction Optimization Procedure 1 (DROP1)	[96]
Decremental Reduction Optimization Procedure 2 (DROP2)	[96]
Decremental Reduction Optimization Procedure 3 (DROP3)	[96]
Patterns by Ordered Projections (POP)	[79]
C-Pruner	[102]

Cuadro 11: Selección basada en eliminación ordenada

Nombre del Algoritmo	Ref.
Random Mutation Hill Climbing (Rmhl)	[91]
Edited Nearest Neighbour by Random Selection (Ennrs)	[97]

emplea como conjunto de entrenamiento para un clasificador preliminar. El objetivo es buscar de entre el conjunto de instancias no clasificadas, aquellas que al clasificarlas pueden añadirse al conjunto de entrenamiento para mejorar o acelerar la clasificación de otras nuevas. El clasificador inicial puede estar seguro de la clasificación efectuada en algunas instancias e inseguro en otras. Las instancias sobre las que está inseguro están situadas en la región de confusión del clasificador. Esta región será mayor cuando el conjunto de entrenamiento es pequeño. El clasificador puede reducir esta región mediante la predicción adecuada de esas instancias donde no hay certeza de su clase.

Conforme se van clasificando nuevas instancias, en caso de que éstas sean lo suficientemente interesantes, serán agregadas al conjunto de entrenamiento para mejorar así las prestaciones que este proporciona.

6. Discretización de Características

Los atributos aparecen en las instancias en muchos casos en diferentes formatos: nominales, discretos y continuos. La necesidad de discretizar los atributos continuos o con un número alto de valores discretos 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 [54, 81, 46].

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 qué puntos combinar se emplea como medida χ^2. La medida χ^2 determina la semejanza de intervalos adyacentes basándose en su nivel de

  relevancia. 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. En la Tabla 12 se muestran los algoritmos que siguen esta estrategia.

- División: La estrategia seguida en este conjunto de técnicas es de ar-

Cuadro 12: Métodos de Discretización por Combinación

Algoritmo	Ref.
ChiMerge	[46]
Chi2	[60]
ConMerge	[94]
USD	[36]

riba 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 crean dichos acumuladores dará lugar a las dos siguientes técnicas basadas en frecuencia y en anchura, según aparecen en la Tabla 13.

Cuadro 13: Métodos de Discretización por División No Supervisados

Algoritmo	Ref.
Igual Anchura	[19]
Igual Frecuencia	[19]

- 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, como la Tabla 14 refleja.

Cuadro 14: Métodos de Discretización por División Supervisados

Algoritmo	Medida	Ref.
ID3 - C4.5	Entropía	[75, 76]
D2	Entropía	[16]
Entropía-MDLP	Entropía	[32]
Contraste	Entropía	[24]
Mantaras	Entropía	[17]
Khiops	Entropía	[8]
Algoritmo de Fayyad e Irani	Entropía	[31]
1R	Acumulación	[41]
Entropía Marginal Máxima	Acumulación	[28]
Zeta	Dependencia	[40]
Cuantización Adaptativa	Acierto	[18]

7. 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 [44]:

- 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 [45]:

- Agrupamiento Jerárquico: Existen una primera división en grupos a nivel mayor, que se van refinando sucesivamente. En la Tabla 15 se presentan los métodos que siguen esta vía.

Cuadro 15: Métodos de Agrupamiento Jerárquico

Algoritmo	Ref.
Conexión Simple	[92]
Conexión Completa	[49]
Mínima Varianza	[70]
Algoritmos Genéticos	[65]

- 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 refleja la Tabla 16:

Cuadro 16: Métodos de Agrupamiento Particional

Algoritmo	Func. a Optimizar	Ref.
K-Medias	Error Cuadrático	[68]
Isodata	Error Cuadrático	[3]
Agrupamiento Dinámico	Error Cuadrático	[93]
Árbol minimal extensivo	Modelo Gráfico	[100]

- 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 [66].

- Agrupamiento difuso: En este caso cada instancia se asocia con cada uno de los grupos empleando una función de pertenencia [99]. La primera aplicación de agrupamiento difuso la encontramos en [83]. El libro de Bezdek es una buena fuente de material sobre el tema de agrupamiento difuso, donde podemos encontrar el algoritmo c-medias difuso [6].

8. Compactación de Datos

Formalmente podemos describir la compactación de datos como un mecanismo de compresión que pretende conservar información estadística [29]. 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 clasicos para efectuar la compactación de los datos y que suelen ser empleados como referencia en las comparativas [29]:

- 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 $w_i = n/p$. 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, reflejados en la Tabla 17.

9. Ejemplo del uso de los Algoritmos Evolutivos para la Selección de Instancias para extraer arboles de decisión

En esta sección presentamos un ejemplo del efecto que produce la reducción de datos como paso previo a la extracción de árboles de decisión. Para ello se muestra el comportamiento de los árboles de decisión sobre los conjuntos de entrenamiento seleccionados mediante algoritmos evolutivos.

Se muestran resultados sobre los árboles extraídos a partir de conjuntos de datos de tamaño mediano (conjunto Satimage con 6435 ejemplos y 36 atributos, disponible en el repositorio de la UCI [69]). En la tabla aparecen reflejados tamaños y porcentajes de clasificación asociados a esos árboles. Como algoritmos consideraremos el C4.5 aplicado directamente sobre el conjunto de datos sin reducir y aplicado sobre el conjunto de datos obtenidos por el algoritmo evolutivo evolutivo CHL para SII ([12, 14]). Ambas evaluaciones han sido llevadas a cabo en un proceso de validación cruzada de orden 10.

En la Tabla 18 se puede apreciar la reducción que se consigue sobre el tamaño del árbol de decisión, con una ligera reducción del porcentaje de acierto. Los modelos resultantes tras el preprocesamiento mejoran la interpretabilidad del modelo, tanto en el número de reglas como en el número de variables por regla.

Otros resultados aplicando algoritmos evolutivos para la selección de instancias se pueden encontrar en [13], donde se estudia su uso conjuntos de datos de gran tamaño para la selección de prototipos, y en [14] donde se analiza la extracción

Cuadro 17: Métodos de Compactación de Datos

Algoritmo	Ref.
Comparación de Momentos sin Depósito	[30]
Modelo basado en Probabilidad	[67]
Comparación empírica de Momentos de Probabilidad	[72]

Cuadro 18: Resultados sobre SatImage

Algoritmo	♯ Reg. Med.	♯ Antec. Med.	Porc. Ac. Test
C4.5	277,5	10,81	86,71 %
CHC	13,8	4,4	84,28 %

de árboles de decisión en conjuntos de datos grandes aplicando algoritmos evolutivos para la selección de conjuntos de entrenamiento.

10. Conclusiones

En este capítulo se presenta la preparación de los datos y su importancia en los procesos de KDD.

En cuanto al uso de los algoritmos evolutivos para la selección de instancias, éstos ofrecen un modelo de reducción de datos prometedor, tanto en precisión como en extracción de modelos interpretables a partir de los conjuntos de datos seleccionados.

En otros ámbitos del KDD los algoritmos evolutivos y las metaheurísticas son igualmente útiles, siendo utilizados en el desarrollo de modelos de extracción de conocimiento (predicción, clasificación, extracción de patrones, ...). En los libros [33, 35] se puede encontrar una perspectiva general sobre el uso de los algoritmos evolutivos en estás áreas.

Acknowledgements

Este estudio está soportado por TIN2005-08386-C05-01 y TIN2005-08386-C05-03.

Referencias

[1] D. W. Aha, D. Kibbler, and M. K. Albert. Instance based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[2] H. Almuallim and T. Dietterich. Learning with many irrelevant features. In *Proceedings of the National Conference on Artificial Intelligence*. AAAI Press/ The MIT Press, 1991.

[3] G.H. Ball and D.J. Hall. Isodata, a novel method of data analisys and classification. Technical report, Stanford University, 1965.

[4] G. Baohua, H. Feifang, and L. Huan. Sampling: Knowing whole from its parts. In H. Liu and H. Motoda, editors, *Instance Selection and Construction for Data Mining*, pages 21–38. Kluwer Academic Publishers, 2001.

[5] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, (5):255–291, 2004.

[6] J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1981.

[7] B. K. Bhattacharya, R. S. Poulsen, and G. T. Toussaint. Application of proximity graphs to editing nearest neighblor decision rule. In *Proceeding of the International Symposium on Information Theory*, 1981.

[8] M. Boulle. Khiops: A statistical discretization method of contin-

uous attributes. *Machine Learning*, 55(1):53–69, 2004.

[9] H. Brighton and C. Mellish. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6:153–172, 2002.

[10] C. E. Broadley. Addresing the selective superiority problem: automatic algorithm/model class selection. In *Proceedings of the Tenth International Machine Learning Conference*, pages 17–24, 1993.

[11] R. M. Cameron-Jones. Instance selection by encoding length heuristic with random mutation hill climbing. In *Proceeding of the Eighth Australian Joint Conference on Artificial Intelligence*, pages 99–106, 1995.

[12] J. R. Cano, F. Herrera, and M. Lozano. Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE Transaction on Evolutionary Computation*, 7(6):561–575, 2003.

[13] J. R. Cano, F. Herrera, and M. Lozano. Stratification for scaling up evolutionary prototype selection. *Pattern Recognition Letters*, 26(7):953–963, 2005.

[14] J.R. Cano, F. Herrera, and M. Lozano. Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability. *Data and Knowledge Engineering*, In press, 2006.

[15] M. Castejón, J. B. Ordieres, F. J. Martínez, and E. P. Vergara. Outlier detection and data cleaning in multivariate non-normal samples: The PAELLA algorithm. *Data Mining and Knowledge Discovery*, (9):171–187, 2004.

[16] J. Catlett. On changing continuous attributes into ordered discrete attributes. In *Proceeding of the Fifth European Working Session on Learning*, pages 164–177. Springer-Verlag, 1991.

[17] J. Cerquides and R. L. Mantaras. Proposal and empirical comparison of a parallelizable distance-based discretization method. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 139–142, 1997.

[18] C.C. Chan, C. Batur, and A. Srinivasan. Determination of quantization intervals in rule based model for dynamic. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, pages 1719–1723, 1991.

[19] M. R. Chmielewski and J. W. Grzymala-Busse. Global discretization of atributes as preprocessing for machine learning. In *Proceeding of the Third International workshop on RSSC94*, pages 294–301, 1994.

[20] D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[21] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. The MIT Press, 1995.

[22] M. Dash. Feature selecion via set cover. In *Proceedings of IEEE Knowledge and Data Engineering Exchange Workshop*, pages 165–171. IEEE Computer Society, 1997.

[23] M. Dash and H. Liu. Hybrid search of feature subsets. In *Pacific Rim International Conference on Artificial Intelligence*, pages 238–249, 1998.

[24] T. Van de Merckt. Decision trees in numerical attribute spaces. *Machine Learning*, pages 1016–1021, 1990.

[25] V. Detours, J.E. Dumont, H. Bersini, and C. Maenhaut. Integration and cross-validation of hihg-throughtput

gene expression data: comparing heterogeneous data sets. *FEBS Letters*, 546(1):98–102, 2003.

[26] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.

[27] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, University of California at Davis, 1992. Tech Report CSE-92-18.

[28] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 194–202, 1995.

[29] W. DuMouchel. *Handbook of Massive Data Sets*, chapter Data squashing: constructing summary data sets, pages 579–591. Kluwer Academic Publishers, 2001.

[30] W. DuMouchel, C. Volinsky, T. Johnson, C. Cortes, and D. Pregibon. Squashing flat files flatter. In *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining*, pages 6–15, 1999.

[31] U. Fayyad and K. Irani. Multi-interval discretization of continuous attributes as preprocessing for classification learning. In *Proc. of the 13th International Join Conference on Artificial Intelligence*, number 1022–1027, 1993.

[32] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceeding of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann, 1993.

[33] A. A. Freitas. *Data mining and knowledge discovery with evolutionary algorithms*. Springer-Verlag, 2002.

[34] G. W. Gates. The reduced nearest neighbour rule. *IEEE Transaction on Information Theory*, 18(5):431–433, 1972.

[35] A. Ghosh and L.C. Jain. *Evolutionary computation in data mining*. Springer-Verlag, 2005.

[36] R. Giráldez, J. Aguilar-Ruiz, J. Riquelme, F. Ferrer-Troyano, and D. Rodríguez. Discretization oriented to decision rules generation. *Frontiers in Artificial Intelligence and Applications*, 82:275–279, 2002.

[37] P. E. Hart. The condensed nearest neighbour rule. *IEEE Transaction on Information Theory*, 18(3):431–433, 1968.

[38] M. Hasenjager and H. Ritter. *New learning paradigms in soft computing*, chapter Active learning in neural networks, pages 137–169. Physica-Verlag GmbH, 2002.

[39] J. Hernández, M.J. Ramirez, and C. Ferri. *Introducción a la Minería de Datos*. Pearson, 2004.

[40] K. M. Ho and P. D. Scott. Zeta: A global method for discretization of continuous variables. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 191–194, 1997.

[41] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–90, 1993.

[42] I. Inza, P. Larrañaga, and B. Sierra. Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms. *International Journal of Approximate Reasoning*, 27(2):143–164, 2001.

[43] I. Inza, M. Merino, P. Larrañaga, J. Quiroga, B. Sierra, and M. Girala. 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, 2001.

[44] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[45] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[46] R. Kerber. Chimerge: discretization of numeric atributes. In *Proceeding of the Ninth National Conference Artificial Intelligence*, pages 123–128. The MIT Press, 1992.

[47] D. Kibbler and D. W. Aha. Learning representative exemplars of concepts: An initial case of study. In *Proc. of the Fourth International Workshop on Machine Learning*, pages 24–30, 1987.

[48] W. Kim, B. Choi, E. Hong, S. Kim, and D. Lee. A taxonomy of dirty data. *Data Mining and Knowledge Discovery*, 7:81–99, 2003.

[49] B. King. Step-wise clustering procedures. *J. Am. Stat. Assoc.*, (69):86–101, 1967.

[50] K. Kira and L. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 249–256, 1992.

[51] P. Krishnaiah and C. Rao. *Handbook of statistics 6: sampling*. North-Holland, 1988.

[52] L. Kuncheva. Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters*, 16:809–814, 1995.

[53] T.Y. Lin. Attribute transformation for data mining I: Theoretical explorations. *International Journal of Intelligent Systems*, 17:213–222, 2002.

[54] H. Liu, F. Hussain, C. Lim Tan, and M. Dash. Discretization: an enabling technique. *Data Mining and Knowledge Discovery*, 6:393–423, 2002.

[55] H. Liu and H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, 1998.

[56] H. Liu and H. Motoda. *Feature selection for knowledge discovery and data mining*. Kluwer Academic Publishers, 1998.

[57] H. Liu and H. Motoda, editors. *Instance selection and construction for data mining*. Kluwer Academic Publishers, 2001.

[58] H. Liu and H. Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6:115–130, 2002.

[59] H. Liu, H. Motoda, and M. Dash. A monotonic measure for optimal feature selection. In *European Conference on Machine Learning*, pages 101–106, 1998.

[60] H. Liu and R. Setiono. Chi2: feature selection and discretization of numeric attributes. In *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391, 1995.

[61] H. Liu and R. Setiono. Feature selection and classification - a probabilistic wrapper approach. In *Proceedings of the Ninth International Conference on Industrial and Ingineering Applications of AI and ES*, pages 419–424, 1996.

[62] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In *Proceedings of International Conference on Machine Learning*, pages 319–327. Morgan Kaufmann Publishers, 1996.

[63] H. Liu and R. Setiono. Incremental feature selection. *Applied Intelligence*, 9(3):217–230, 1998.

[64] D. G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7(1):72–85, 1995.

[65] J.A. Lozano and P. Larrañaga. Applying genetic algorithms to search for the best hierarchical clustering of a data set. *Pattern Recognition Letters*, 20(9):911–918, 1999.

[66] S.Y. Lu and K.S. Fu. A sentence-to-sentence clustering procedure for pattern analisys. *IEEE Trans. Syst. on Man Cybernetics*, (8):381–389, 1978.

[67] D. Madigan, N. Raghavan, W. Domouchel, M.Ñason, C. Posse, and G. Ridgeway. Likelihood-based data squashing: a modeling approach to instance construction. *Data Mining and Knowledge Discovery*, 6:173–190, 2002.

[68] J. McQueen. Some methods for classification and analisys of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistic and Probability*, pages 281–287, 1967.

[69] C. J. Merz and P. M. Murphy. UCI repository of machine learning databases. 1996. University of California Irvine, Department of Information and Computer Science, http://kdd.ics.uci.edu.

[70] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms which use cluster centers. *Comput. J.*, (26):354–359, 1984.

[71] P.Ñarendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transaction on Computers*, C-26(9):917–922, 1977.

[72] A. B. Owen. Data squashing by empirical likelihood. *Data Mining and Knowledge Discovery*, 7(1):101–113, 2003.

[73] F. J. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Knowledge Discovery and Data Mining*, pages 23–32, 1999.

[74] D. Pyle. *Data preparation for data mining*. Morgan Kaufmann, 1999.

[75] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[76] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, 1993.

[77] T. Ravindra and M.Ñarasimha. Comparison of genetic algorithm based prototype selection schemes. *Pattern Recognition*, 34:523–525, 2001.

[78] C. R. Reeves and D. R. Bush. *Instance Selection and Construction for Data Mining*, chapter Using genetic algorithms for training data selection in RBF networks, pages 339–356. Kluwer Academic Publishers, 2001.

[79] J.C. Riquelme, J.S. Aguilar, and M. Toro. Finding representative patterns with ordered projections. *Pattern Recognition*, 36(4):1009–1018, 2003.

[80] G. L. Ritter, H. B. Woodruff, S. R. Lowry, and T. L. Isenhour. An algorithm for a selective nearest neighbour decision rule. *IEEE Transaction on Information Theory*, 21(6):665–669, 1975.

[81] J. Aguilar Ruiz, J. Bacardit, and F. Divina. Experimental evaluation of discretization schemes for rule induction. In *Proc. of the Genetic Evolutionary Computation- GECCO 2004, Genetic and Evolutionary Computation Conference*, pages 828–839, 2004.

[82] R. Ruiz, J.C. Riquelme, and J.S. Aguilar. Projection-based measure for efficient feature selection. *Journal of Intelligent and Fuzzy Systems*, 12(3-4):175–183, 2002.

[83] E.H. Ruspini. A new approach to clustering. *Inf. Control*, (15):22–32, 1969.

[84] M. Saar-Tsechansky. Active sampling for class probability estimation and ranking. *Machine Learning*, 54:153–178, 2004.

[85] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning, 2002.

[86] E. Schallehn, K. Sattler, and G. Saake. Efficient similarity-based operations for data integration. *Data and Knowledge Engineering*, (48):361–387, 2004.

[87] R. Scheaffer, W. Mendenhall, and L. Ott. *Elementary survey sampling*. Duxbury Press, 1996.

[88] M. Sebban, R.Ñock, J. H. Chauchat, and R. Rakotomalala. Impact of learning set quality and size on decision tree perfomances. *International Journal of Computers, Systems and Signals*, 1(1):85–105, 2000.

[89] H. Shinn-Ying, L. Chia-Cheng, and L. Soundy. Design of an optimal nearest neighbour classifier using an intelligent genetic algorithm. *Pattern Recognition Letters*, 23(13):1495–1503, 2002.

[90] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[91] D. B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *International Conference on Machine Learning*, pages 293–301, 1994.

[92] P.H.A. Sneath and R.R. Sokal. *Numerical Taxonomy*. Freeman, 1973.

[93] M.J. Symon. Clustering criterion and multi-variate normal mixture. *Biometrics*, (77):35–43, 1977.

[94] K. Wang and B. Liu. Concurrent discretization of multiple attributes. In *Pacific-Rim International Conference on AI*, pages 250–259, 1998.

[95] D. L. Wilson. Asymptotic properties of nearest neighbour rules using edited data. *IEEE Transaction on Systems, Man. and Cybernetics*, 2:408–420, 1972.

[96] D. R. Wilson and T. R. Martinez. Instance pruning techniques. In *Proceedings of the International Conference*, pages 403–411, 1997.

[97] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–268, 2000.

[98] L. Xu, P. Yan, and T. Chang. Best first strategy for feature selection. In *Proceedings of the Ninth International Conference on Pattern Recognition*, pages 706–708, 1988.

[99] L.A. Zadeh. Fuzzy sets. *Inf. Control*, (8):338–353, 1965.

[100] C.T. Zhan. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Computation*, C-20:68–86, 1971.

[101] S. Zhang, C. Zhang, and Q. Yang. Data preparation for data mining. *Applied Artificial Intelligence*, 17:375–381, 2003.

[102] K. P. Zhao, S. G. Zhou, J. H. Guan, and A. Y. Zhou. C-pruner: An improved instance prunning algorithm. In *Proceeding of the Second International Conference on Machine Learning and Cybernetics*, pages 94–99, 2003.