Extracción de modelos predictivos e interpretables en conjuntos de datos de tamaño grande mediante la selección de conjuntos de entrenamiento

José Ramón Cano, Francisco Herrera y Manuel Lozano

Dept. de Informática
EUP de Linares
Univ. de Jaén
23700 Linares (Jaén)
jrcano@ujaen.es

Dept. de Ciencias de la Computación
e Inteligencia Artificial
ETS Ingeniería Informática
Univ. de Granada
18071 Granada
herrera@decsai.ugr.es, lozano@decsai.ugr.es

Resumen

La extracción de modelos predictivos es una tarea frecuente en Minería de Datos y tiene como objetivo la generación de modelos precisos e interpretables. La reducción de datos es un preprocesamiento interesante que puede ser empleado para extraer modelos con estas características en conjuntos de gran tamaño. En este estudio analizamos la extracción de modelos predictivos basados en reglas a partir de los conjuntos de entrenamiento obtenidos mediante un algoritmo de selección de instancias evolutivo estratificado. Este método permite enfrentarse al problema de escalado que aparece al evaluar conjuntos de datos de gran tamaño, ofreciendo modelos precisos e interpretables. [1]

1. Introducción

En Minería de Datos (MDD) la extracción de modelos representativos es un proceso básico [21]. Los modelos, dependiendo del ámbito en el que se vayan a emplear pueden ser:

- Modelos Predictivos. El objetivo perseguido en estos modelos es la precisión. En la literatura podemos encontrar diferentes propuestas para evaluar la calidad de estos modelos, tales como su simplicidad, interpretabilidad, etc [14].
- Modelos Descriptivos. Este tipo de modelos intentan encontrar relaciones y patrones de comportamiento en el conjunto de datos para ofrecer conocimiento sobre un problema concreto de MDD [9].

En este estudio centramos la atención en los modelos predictivos basados en reglas de clasificación. Los modelos, concretamente árboles de decisión, son extraídos a partir de los conjuntos de datos utilizando el algoritmo C4.5 de Quinlan [16]. Para analizar la calidad de los árboles generados se pueden emplear diferentes medidas [14].

Los modelos son generados empleando como entrada conjuntos de datos de tamaño grande. Debido a ello, los modelos extraídos presentan tamaños elevados, lo cual disminuye su interpretabilidad.

Con objeto de mejorar dicha interpretabilidad, los conjuntos de instancias son preprocesados mediante algoritmos de reducción de datos. Se pretende con ello seleccionar del conjunto original, conjuntos de entrenamiento tales que permitan extraer modelos con equilibrio entre interpretabilidad y precisión [18]. La selección de conjuntos de entrenamiento se

[1] Este trabajo esta soportado por la Comisión Interministerial de Ciencia y Tecnología con el proyecto TIC2002-04036-C05-01

puede llevar cabo utilizando algoritmos de selección de instancias (SII), siguiendo una determinada estrategia [12, 20].

Los algoritmos evolutivos (AAEE [7]) son métodos adaptativos basados en la evolución natural que pueden ser aplicados a problemas de optimización tales como la SII [19]. En este estudio emplearemos el algoritmo CHC ([4]) de entre los AAEE considerando el comportamiento que muestra en [2].

Los algoritmos de SII aplicados sobre conjuntos de datos grandes pueden ser ineficaces e ineficientes. Al efecto que produce el tamaño del conjunto de datos sobre los algoritmos lo llamaremos problema de escalado.

Este estudio analiza la SII evolutiva de conjuntos de entrenamiento sobre conjuntos de tamaño grande con el objetivo de obtener reglas con índices elevados de interpretabilidad y precisión. Para hacer frente al problema de escalado se combina la estratificación de los conjuntos de datos con la SII evolutiva sobre ellos. Mediante la estratificación reducimos el tamaño del conjunto original, dividiéndolo en estratos en los que la selección será aplicada. Se ha analizado la calidad de los conjuntos de entrenamiento seleccionados a través de los modelos (árboles de decisión) que extraemos a partir de ellos desde las perspectivas de la precisión y la interpretabilidad.

Este estudio está organizado de la siguiente manera. En la Sección 2 se analizan los modelos predictivos y las medidas consideradas para evaluar su comportamiento. La Sección 3 describe el problema de escalado debido a la evaluación de conjuntos de gran tamaño y como afecta a los algoritmos de selección de instancias y de extracción de reglas. La Sección 4 presenta la selección evolutiva estratificada de conjuntos de entrenamiento. La Sección 5 contiene el estudio experimental desarrollado, ofreciendo la metodología seguida, los resultados y su análisis. Finálmente, en la Sección 6 se muestran las conclusiones alcanzadas.

2. Modelos predictivos: extracción de árboles de decisión con C4.5

Uno de los principales beneficios que ofrece la utilización de árboles de decisión y reglas es que permiten obtener modelos comprensibles, un punto clave para su utilidad y su aplicación. Concretamente, en este estudio los árboles de decisión han sido generados empleando el algoritmo C4.5 [16]. Los modelos generados son completos y consistentes, cubriendo totálmente el conjunto de ejemplos. Dicha situación origina que los modelos se sobreajusten con respecto al conjunto de entrenamiento y disminuyan su precisión al clasificar nuevos ejemplos. Como añadido, estos modelos son sensibles a la presencia de ruido en el conjunto de entrenamiento, ajustando sus ramas y nodos a él. Para limitar estas desventajas, se aplican mecanismos de poda a los árboles generados [5]:

- Métodos de prepoda. El proceso de poda se desarrolla durante la generación del árbol. La poda determina el criterio de parada en la especialización de las ramas.
- Métodos de postpoda. En este caso, el proceso de poda se aplica después de la construcción del árbol. La poda elimina nodos de abajo hacia arriba del árbol hasta que se alcanza un determinado límite.

Los mecanismos de poda aumentan la capacidad de generalización del modelo y reducen su tamaño, lo cual mejora su interpretabilidad.

El inconveniente de ambos mecanismos de poda es determinar el límite de parada. Este límite dependerá del conjunto de entrenamiento del que se extraiga el árbol de decisión. El ajuste adecuado de este límite produce modelos con mejores o peores prestaciones. Si la poda es minimal, el sobreajuste se mantiene. Si la poda es máxima, la precisión del modelo podría reducirse debido a una excesiva generalización.

En el caso del algoritmo C4.5, se emplea la poda basada en el error [16]. Esta estrategia de poda presenta un mayor equilibrio entre la precisión y el tamaño del árbol que otros mecanismos de poda [5].

Cuando el árbol de decisión se emplea en dominios donde su carácter predictivo y descriptivo es importante, la simplicidad del árbol de decisión es un factor clave [11]. Las medidas que se emplean en este estudio para evaluar los modelos predictivos generados con C4.5 son las siguientes [14]:

Precisión en test. En el aprendizaje de modelos predictivos, el optimizar la precisión del conjunto de reglas es un factor clave. El modelo se genera empleando como entrada el conjunto de entrenamiento seleccionado. El porcentaje de test se calcula utilizando el modelo construido (lo denotaremos como TEST).

Tamaño del árbol de decisión. El tamaño del árbol se evalúa mediante el número de reglas (n_R) que componen el modelo ($TAM = n_R$).

Numero de antecedentes. Como segunda medida del tamaño del árbol de decisión se utiliza el número medio de antecedentes por regla. Considerando las reglas R_i de la forma $Cond \rightarrow Clase$, $N_{Antec}(R_i)$ es el número de antecedentes de R_i y ANT el número medio de antecedentes del modelo: $ANT = \frac{1}{n_R} \sum_{i=1}^{n_R} N_{Antec}(R_i)$.

Tanto el número de reglas (TAM) como el número de medio de antecedentes (ANT) son empleados para analizar la interpretabilidad del modelo.

3. El problema de escalado

En esta sección se estudia el efecto del tamaño del conjunto de datos sobre los algoritmos de SII y en la extracción de árboles de decisión.

La mayoría de los algoritmos de SII presentan problemas al ser aplicados sobre conjuntos de datos grandes. Deben hacer frente principalmente a las siguientes dificultades:

Eficiencia. La eficiencia de los algoritmos de SII no evolutivos es al menos de $O(n^2)$, siendo n el número de instancias en el conjunto de datos. Existe otro conjunto de algoritmos (como el Rnn [6], Shrink [10], etc.) pero la mayoría presenta un orden de eficiencia muy superior a $O(n^2)$. Conforme el tamaño del conjunto crece, el tiempo necesario para evaluar cada algoritmo aumenta también.

Recursos. La mayoría de los algoritmos evaluados necesitan tener almacenado el conjunto completo de datos en memoria para poder ejecutarse. Si el tamaño de este conjunto es demasiado grande, podría ser necesario el empleo del disco duro como memoria de intercambio. Esta situación supone un efecto adverso en la eficiencia debido al excesivo acceso a disco.

Representación. Los AAEE sufren el efecto del tamaño del conjunto de datos en la representación de las soluciones (tamaño de los cromosomas). Cuando el tamaño de estos cromosomas es demasiado grande, el algoritmo ve afectada su convergencia así como su tiempo de ejecución.

Estas desventajas producen una degradación considerable del comportamiento de los algoritmos de SII. La evaluación de los algoritmos directamente sobre el conjunto de datos completo puede ser ineficaz e ineficiente.

A su vez, el tamaño del árbol de decisión generado empleando conjuntos de datos de tamaño grande se ve aumentado considerablemente [15]. El crecimiento del tamaño del árbol provoca:

Sobreajuste. En este caso, la hipótesis aprendida se encuentra demasiado relacionada con los conjuntos de entrenamiento, lo que origina que su capacidad de generalización se vea penalizada [17].

Pérdida de interpretabilidad. El elevado tamaño de los árboles de decisión produce una excesiva complejidad en ellos, que puede provocar el que sean incomprensibles para los expertos [18, 22].

4. Selección de conjuntos de entrenamiento evolutiva estratificada

Para llevar a cabo la SII en conjuntos de gran tamaño se ha combinado la estratificación del conjunto inicial con los AAEE. Siguiendo esta vía, el algoritmo puede ser aplicado independientemente del tamaño del conjunto. La estratificación reduce el espacio de búsqueda, al mismo tiempo que el algoritmo evolutivo explora cada estrato.

La Subsección 4.1 describe el proceso de selección de conjuntos de entrenamiento. En la Subsección 4.2 se muestra el empleo de los algoritmos evolutivos en la selección de conjuntos de entrenamiento. Finálmente, la Subsección 4.3 está dedicada a presentar la selección de conjuntos de entrenamiento evolutiva estratificada.

4.1. Selección de conjuntos de entrenamiento

El objetivo de esta selección consiste en encontrar el conjunto de entrenamiento del que se pueden extraer, cuando es empleado como entrada, conjuntos de reglas con precisión e interpretabilidad altas.

En selección de conjuntos de entrenamiento, el conjunto inicial (D) se divide en conjunto de entrenamiento (TR) y test (TS). Empleando TR como entrada, el algoritmo de selección de instancias obtiene el conjunto de entrenamiento seleccionado (TSS). Este conjunto es utilizado por el algoritmo C4.5 para extraer el modelo a partir de él. Finálmente, el modelo se valida empleando el conjunto de test TS.

4.2. Algoritmos evolutivos para selección de conjuntos de entrenamiento

Para describir el empleo de los AAEE en SII hay que referenciar dos factores clave: la especificación de la representación y la definición de la función objetivo.

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.

Función objetivo: Se han empleado dos funciones objetivo distintas. Sea TSS un subconjunto de instancias de TR codificadas en un cromosoma para ser evaluado. Definiremos las funciones de evaluación como combinación de dos valores:

- El porcentaje de clasificación con el vecino más cercano asociado a TSS ($porc\_clas_1$) y el porcentaje de reducción conseguido en TSS con respecto a TR ($porc\_red_1$):

$$F_{Eval1}(TSS) = \alpha \cdot porc\_clas_1 + (1-\alpha) \cdot porc\_red_1. \quad (1)$$

  Para calcular el porcentaje de clasificación ($porc\_clas_1$) se emplea el clasificador 1-vecino más cercano. 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 $TSS \setminus \{y\}$.

  El porcentaje de reducción ($porc\_red_1$) se obtiene de la siguiente forma:

$$porc\_red_1 = \frac{100 \cdot (|TR| - |TSS|)}{|TR|} \quad (2)$$

- El porcentaje de clasificación con C4.5 asociado a TSS ($porc\_clas_2$) y el porcentaje de reducción conseguido en el tamaño del modelo extraído de TSS con respecto al extraído de TR ($porc\_red_2$):

$$F_{Eval2}(TSS) = \alpha \cdot porc\_clas_2 + (1-\alpha) \cdot porc\_red_2. \quad (3)$$

  Para calcular el porcentaje de clasificación se emplea el modelo generado por C4.5 a partir de TSS ($porc\_clas_2$). Dicho porcentaje representa el porcentaje de muestras clasificadas correctamente de TR empleando tan solo instancias de TSS para confeccionar el árbol de decisión.

  El porcentaje de reducción ($porc\_red_2$) se obtiene de la siguiente forma:

$$porc\_red_2 = \frac{100 \cdot (TAM_{TR} - TAM_{TSS})}{TAM_{TR}} \quad (4)$$

Para diferenciar el empleo de una función objetivo u otra en el algoritmo CHC se utilizara la notación CHC_1 para la primera función objetivo y CHC_2 para la segunda.

4.3. Selección de conjuntos de entrenamiento evolutiva estratificada

La estratificación ha sido empleada en trabajos previos para hacer frente al problema de escalado ofreciendo buenos resultados [3]. Consiste en dividir el conjunto inicial en subconjuntos disjuntos manteniendo la distribución de las clases.

El número de estratos determinará el tamaño de cada uno de ellos. Empleando el número adecuado de estratos se puede reducir sensiblemente el tamaño del conjunto de datos, reduciéndose el problema de escalado.

Con la estratificación, el conjunto D se divide en t subconjuntos disjuntos D_j, de igual tamaño. El conjunto de entrenamiento TR se obtiene como muestra la expresión (5) y TS, su complementario en D, como indica la expresión (6).

$$TR = \bigcup_{j \in J} D_j, J \subset \{1, 2, ..., t\} \quad (5)$$

$$TS = D \setminus TR \quad (6)$$

Los algoritmos de SII se aplican a cada D_j, obteniendo los conjuntos seleccionados DS_j. El conjunto de entrenamiento seleccionado (TSS) siguiendo la estrategia de estratificación se obtendrá empleando los DS_j (ver expresión (7)) y se denomina subconjunto estratificado de entrenamiento (STSS).

$$STSS = \bigcup_{j \in J} DS_j, J \subset \{1, 2, ..., t\} \quad (7)$$

5. Estudio experimental

En esta sección se describe el estudio experimental desarrollado. La Subsección 5.1 presenta la metodología seguida en los experimentos. La Subsección 5.2 muestra los resultados que, finálmente en la Subsección 5.3, se analizan.

5.1. Metodología de experimentación

En esta subsección se presentan: los conjuntos de datos, algoritmos y parámetros considerados, el esquema de estratificación y la extracción de modelos seguida en el algoritmo C4.5.

5.1.1. Conjuntos de datos, algoritmos de selección de instancias y parámetros

Los experimentos se han llevado a cabo aumentando el tamaño y la complejidad de los conjuntos de datos. Se han seleccionado conjuntos de tamaño grande y muy grande, como podemos ver en la Tabla 1 (los conjuntos han sido seleccionados del depósito de UCI [13], donde el conjunto Kdd Cup'99 corresponde a su versión al 10 %).

Cuadro 1: Conjuntos de Datos

Conjunto	Inst.	Atrib.	Clases
Adult	30132	14	2
Kdd Cup'99	494022	41	23

Los algoritmos evaluados se han dividido en dos grupos, considerando su naturaleza evolutiva:

Algoritmos no evolutivos. Los algoritmos escogidos son: Cnn [8], Ib2 [1] y Ib3 [1]. Han sido seleccionados por ser los que han mostrado ser mas eficientes en [2]. Los parámetros de Ib3 son: Nivel de Aceptación=0,9 y Nivel de Eliminación=0,7. Los otros algoritmos no presentan parámetros que necesiten ajuste.

Algoritmos evolutivos. Se ha seleccionado el algoritmo CHC como algoritmo eficaz y eficiente por su comportamiento en [2]. Se ha empleado un tamaño de población de 50 cromosomas y 10000 evaluaciones. El valor de α en ambas funciones objetivo es 0,5 según [2, 3].

5.1.2. Estratificación y particiones

Cada algoritmo ha sido ejecutado siguiendo un proceso de validación cruzada de orden 10. De esta forma, TR_i, con i=1,...,10 es un 90 % del

conjunto D y TS_i su complementario al 10 % en D.

A la ejecución siguiendo el proceso de estratificación la llamaremos validación cruzada estratificada de orden 10 (Tfcv st).

En Tfcv st cada TR_i y TS_i se define como indican las expresiones (8) y (9), mediante la unión de subconjuntos D_j.

$$TR_i = \bigcup_{j \in J} D_j, \quad J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \tag{8}$$

$$TS_i = D \setminus TR_i \tag{9}$$

donde t es el número de estratos, y b el número de estratos agrupados ($b = t/10$ para llevar a cabo la validación de orden 10).

El conjunto $STSS_i$ es generado empleando los subconjuntos DS_j seleccionados en vez de los D_j (ver exprexión (10)).

$$STSS_i = \bigcup_{j \in J} DS_j, \quad J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \tag{10}$$

Los conjuntos de datos han sido divididos en 100 estratos (t=100).

5.1.3. Evaluación de C4.5

Como referencia incorporamos la evaluación del algoritmo C4.5 sobre el conjunto de datos inicial sin reducción, empleando una validación cruzada de orden 10 sin estratificar (la llamaremos Tfcv cl).

Se ha incluido al mismo tiempo la ejecución de C4.5 aplicando poda máxima (C4.5 Max) y mínima (C4.5 Min) y la que emplea por defecto (C4.5). En todos los casos se trata de la poda basada en el error. El objetivo es analizar el efecto de la poda en el árbol desde la perspectiva de la precisión y la interpretabilidad.

5.2. Resultados

La tabla de resultados presenta la siguiente estructura: La primera columna muestra el nombre del algoritmo. En esta columna el nombre es seguido por el tipo de validación empleada, *st* (Tfcv st) o *cl* (Tfcv cl). La segunda columna ofrece el porcentaje de reducción media conseguido por el algoritmo de selección sobre el conjunto inicial de datos. La tercera columna contiene el porcentaje de acierto en test del árbol de decisión generado a partir del subconjunto seleccionado. La cuarta columna presenta el número medio de reglas que componen el modelo. La quinta columna muestra el número medio de antecedentes de las reglas que componen el modelo.

A continuación se presentan las tablas de resultados obtenidas:

Cuadro 2: Resultados en datos Adult

	RED	TEST	TAM	ANT
C4.5 Min cl		84,02	1252	17,31
C4.5 cl		85,4	359	14,38
C4.5 Max cl		85,86	52	11,15
Cnn st	84,27	85,75	292	15,56
Ib2 st	99,57	36,4	12	5,08
Ib3 st	76,69	82,7	179	12,86
CHC_1 st	99,38	82,7	5	2,80
CHC_2 st	58,54	85,24	216	14,51

Cuadro 3: Resultados en Kdd Cup'99

	RED	TEST	TAM	ANT
C4.5 Min cl		99,96	281	15,07
C4.5 cl		99,95	143	11,78
C4.5 Max cl		99,99	106	10,44
Cnn st	63,85	99,5	105	12,11
Ib2 st	82,01	95,05	58	10,86
Ib3 st	78,82	96,77	74	11,48
CHC_1 st	99,28	98,41	9	3,56
CHC_2 st	60,32	99,7	69	10,03

5.3. Análisis

El análisis de las Tablas 2 y 3 permite hacer las siguientes observaciones:

- Considerando el porcentaje de reducción sobre el conjunto original de muestras, la selección evolutiva estratificada (en su versión CHC_1) presenta el mejor comportamiento de entre los algoritmos analizados, conforme crece el tamaño del conjunto.

- El tamaño del modelo predictivo depende del tamaño del conjunto de entrenamiento que se emplea como entrada para generarlo. De esta forma, los algoritmos de selección que presentan los mayores porcentajes de reducción son los tienen asociados los árboles de decisión más pequeños. El algoritmo de selección evolutivo estratificado, en su versión CHC_1, consigue porcentajes de reducción de los más elevados, junto con el algoritmo Ib2, sin embargo los modelos predictivos extraídos a partir del conjunto seleccionado por el CHC_1 son mucho menores.

- Del empleo de las dos funciones objetivo (CHC_1 y CHC_2) habría que destacar que en CHC_1 existe una mayor tendencia a reducir el conjunto inicial, lo que supone menores modelos asociados y un leve descenso en la precisión. En CHC_2 la reducción es menor, con lo lo cual los modelos extraídos tienen un tamaño superior que los ofrecidos por CHC_1 junto con una precisión cercana a la de C4.5.

La selección evolutiva estratificada consigue modelos con precisiones cercanas a las ofrecidas por C4.5 sin reducción, con el añadido de presentar unos modelos con tamaños muy inferiores a los que se consigue con C4.5, aún llevando a cabo poda máxima. Así mismo, habría que destacar el comportamiento de CHC_1. Éste ofrece reducciones superiores al 99% del conjunto original, con porcentajes en precisión tan solo un 3% menores que las ofrecidas por C4.5 sin reducción. La reducción del tamaño del modelo es más que significativa al compararlo con los demás (por ejemplo, en Kdd Cup'99 de 143 reglas y 14,38 antecedentes obtenidos por C4.5 con poda por defecto, a las 9 reglas y 3,56 antecedentes por regla de CHC_1).

6. Conclusiones

Es este estudio se ha analizado la extracción de modelos predictivos basados en reglas mediante la selección de conjuntos de entrenamiento sobre conjuntos de datos de tamaño grande. La calidad de los modelos ha sido evaluada desde la perspectiva de su precisión y de su interpretabilidad. Las principales conclusiones alcanzadas son las siguientes:

- La selección evolutiva estratificada, en su versión CHC_1 ofrece los porcentajes de reducción más altos del estudio conforme se incrementa el tamaño del conjunto.

- El algoritmo CHC estratificado, en su versión CHC_2 muestra los mejores porcentajes de acierto de entre los algoritmos de selección conforme crece el tamaño del conjunto, solo superados por el algoritmo C4.5 sin reducir.

- Prestándo atención al tamaño de los modelos se puede destacar que el algoritmo CHC_1 estratificado ofrece los árboles de decisión compuestos por el menor número de reglas y de antecedentes, siendo los más interpretables.

Como conclusión final consideramos que la extracción de modelos predictivos mediante la selección evolutiva estratificada presenta el mejor comportamiento de entre los algoritmos de selección de instancias analizados. Ofrece los árboles de decisión más pequeños con indices de precisión altos, similares a los que muestra el C4.5 sin reducción. La selección evolutiva estratificada permite obtener modelos predictivos con el equilibrio adecuado entre interpretabilidad y precisión.

Referencias

[1] Aha, D. W., Kibbler, D., y Albert, M. K., Instance-Based learning algorithms, *Machine Learning*, 6, 37–66, 1991.

[2] Cano, J. R., Herrera, F., y Lozano, M. Using evolutionary algorithms as instance selection for data reduction in KDD: An

experimental study, *IEEE Transaction on Evolutionary Computation*, 7 (6), 561–575, 2003.

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

[4] Eshelman, L. J., The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination, *Foundations of Genetic Algorithms*, 1, 265–283, 1991.

[5] Esposito, F., Malerba, D., y Semeraro, G., A comparative analysis of methods for pruning decision trees, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (5), 476–491, 1997.

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

[7] Goldberg, D. E., *The design of competent genetic algorithms: Steps toward a computational theory of innovation*, Kluwer Academic Pub., 2002.

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

[9] Hipp, J., Guntzer, U., y Nakhaeizadeh, G., Algorithms for Association Rule Mining: A General Survey and Comparison, *SIGKDD Explorations*, 2 (1), 58–64, 2000.

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

[11] Last, M., y Maimon; O., A compact and accurate model for classification, *IEEE Transactions on Knowledge and Data Engineering*, 16 (2), 203–215, 2004.

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

[13] Merz, C. J., y Murphy, P. M., *UCI repository of machine learning databases*, University of California Irvine, Department of Information and Computer Science, 1996.

[14] Kweku-Muata y Osei-Bryson, Evaluation of decision trees: a multicriteria approach, *Computers and Operations Research*, 31, 1933–1945, 2004.

[15] Oates, T., y Jensen, D., Large datasets lead to overly complex models: an explanation and a solution, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 294–298, 1998.

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

[17] Schaffer, C., When does overfitting decrease prediction accuracy in induced decision trees and rule sets?, In *Proceedings of the European Working Session on Learning (EWSL-91)*, 192–205, 1991.

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

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

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

[21] Witten, I. H., y Frank, E., *Data mining: practical machine learning tools and techniques with java implementations*, Morgan Kaufmann, 2000.

[22] Zhou, Z.-H., y Jiang, Y., Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble, *IEEE Transactions on Information Technology in Biomedicine*, 7 (1), 37–42, 2003.