Seleccion Evolutiva Estratificada de Conjuntos de Entrenamiento para la Obtención de Bases de Reglas con un Alto Equilibrio entre Precisión e Interpretabilidad

José Ramón Cano[1], Francisco Herrera[2] y Manuel Lozano[2]

[1] Universidad de Jaén, Departamento de Informática,
Escuela Politécnica Superior de Linares, 23700, Linares, Jaén, España.
jrcano@ujaen.es
[2] Universidad de Granada, Departamento de Ciencias de la Computación e Inteligencia Artificial, Escuela Técnica Superior de Ingeniería Informática,
18071, Granada, España
herrera@decsai.ugr.es, lozano@decsai.ugr.es

Resumen La generación de modelos predictivos es un tema de interés en Minería de Datos. El objetivo es la generación de modelos con alta precisión e interpretabilidad. La selección de conjuntos de entrenamiento mediante la selección de prototipos es un camino prometedor para obtener modelos predictivos con estas características. En este trabajo, analizamos la extracción de modelos predictivos basados en reglas a partir de la selección evolutiva de conjuntos de entrenamiento estratificada. Esta técnica permite abordar el problema de escalado que puede aparecer cuando se manejan conjuntos de entrenamiento de gran tamaño. [3]

1. Introducción

Un proceso básico en *Minería de Datos* (MDD [1]) es la extracción de modelos. Los modelos, dependiendo del dominio sobre el que se aplican, pueden ser:

- *Modelos Predictivos*. La finalidad perseguida por estos modelos es la clasificación o precisión del modelo. En la actualidad, existen diferentes propuestas para medir la calidad de los modelos predictivos que incluyen la interpretabilidad, simplicidad, etc. [2].
- *Modelos Descriptivos*. Con ellos, se pretende encontrar relaciones o patrones de comportamiento que aporten conocimiento sobre el problema.

En este capítulo centraremos nuestra atención en los modelos predictivos basados en reglas de clasificación. En particular, los analizamos considerando la precisión e interpretabilidad que ofrecen. La extracción de los modelos la

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

llevamos a cabo empleando el algoritmo `C4.5` aplicado sobre los conjuntos de datos [3]. En [2] podemos encontrar diferentes medidas para analizar la calidad de los árboles obtenidos.

Una posible vía para mejorar las prestaciones de los modelos predictivos (precisión e interpretabilidad) es utilizar conjuntos de entrenamiento adecuados. La selección de los conjuntos de entrenamiento se puede llevar a cabo mediante algoritmos de selección de prototipos. Los algoritmos de selección de prototipos escogen un conjunto de instancias representativas de acuerdo a un criterio de selección, y pueden mejorar la capacidad de predicción del clasificador según la regla del vecino más cercano [4,5].

Los *Algoritmos Evolutivos* (`AAEE`) son métodos adaptables basados en la evolución natural que pueden ser utilizados para problemas de búsqueda y optimización [6]. Los `AAEE` ofrecen resultados interesantes cuando se emplean en la selección de prototipos [7,8]. En este estudio, emplearemos el algoritmo evolutivo `CHC` [9]. Éste ha demostrado un excelente comportamiento como selector evolutivo de prototipos en el estudio presentado en [10].

Nos centramos en la selección de prototipos para conjuntos de gran tamaño. Dicha circunstancia puede suponer que al aplicar los algoritmos de selección de prototipos directamente sobre ellos, éstos sean ineficientes e ineficaces. El efecto que produce el tamaño del conjunto de datos sobre los algoritmos extracción de conocimiento se denomina problema de escalado.

Para abordar el problema de escalado combinamos la estratificación de los conjuntos de datos con la selección evolutiva sobre ellos. La estratificación reduce el tamaño del conjunto original dividiéndolo en porciones sobre las cuales se efectuarán las selecciones. Analizamos los conjuntos de entrenamiento a partir de los modelos (árboles de decisión) obtenidos a partir de ellos, desde la doble perspectiva de precisión e interpretabilidad.

El capítulo se organiza en las siguientes secciones. En la Sección 2, se presentan los modelos predictivos y las medidas de calidad consideradas para evaluarlos. El problema de escalado que sufren los algoritmos de selección de prototipos y la extracción de árboles de decisión se analiza en la Sección 3. La Sección 4 presenta la selección evolutiva estratificada de prototipos aplicada a selección de conjuntos de entrenamiento para la obtención de modelos predictivos. La Sección 5 muestra el estudio experimental desarrollado, acompañado de la metodología seguida, resultados y análisis de los mismos. Finalmente en la Sección 6 se alcanzan las conclusiones.

2. Modelos Predictivos: Árboles de Decisión para Clasificación Extraídos con C4.5

La importancia de los árboles de decisión se debe a su capacidad de construir modelos interpretables, siendo éste un factor decisivo para su aplicación. La clasificación en árboles de decisión considera clases disjuntas, de forma que el árbol conducirá a una y solo una hoja, asignando una única clase a la predicción.

En nuestro caso vamos a emplear el algoritmo `C4.5` para generar los árboles de decisión [3]. El algoritmo `C4.5` es un algoritmo de *partición* o de tipo "divide y vencerás".

Los modelos generados 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. 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. Podemos distinguir dos tipos de poda:

- *Prepoda*: El proceso se lleva a cabo durante la construcción del árbol. Se trata de determinar el criterio de parada a la hora de seguir especializando una rama o regla.
- *Postpoda*: El proceso se lleva a cabo después de la construcción del árbol. Se trata de eliminar nodos de abajo a arriba del árbol hasta un cierto límite.

El algoritmo `C4.5` se caracteriza por combinar prepoda y postpoda. El mecanismo de poda, además de mejorar la capacidad de generalización del modelo, reduce su tamaño, lo que aumenta su interpretabilidad.

El inconveniente, tanto de los mecanismos de prepoda como de los de postpoda, es que hay que indicar el criterio de parada en uno y el limite de eliminación respectivamente. Dichos factores van a depender del conjunto de entrenamiento en cuestión y dependiendo de su adecuado ajuste pueden producir modelos con mayores o menores prestaciones. Si la poda es mínima seguirá manteniéndose el ajuste sobre los datos de entrenamiento. Si la poda es máxima podría perderse capacidad de predicción al generalizar en exceso.

En las situaciones en las que el árbol de decisión va a ser utilizado de forma predictiva y descriptiva, es importante que el árbol de decisión sea tan simple como sea posible [2]. Las medidas que emplearemos para evaluar las prestaciones de los modelos predictivos extraídos por el algoritmo `C4.5` serán por tanto las siguientes:

2.1. Porcentaje de Acierto en Test

En el aprendizaje de modelos predictivos se pretende maximizar la precisión del conjunto de reglas inducido. Por tanto, hay que considerar como medida de calidad el porcentaje de acierto en test que el modelo extraído nos proporciona. El modelo se genera mediante el algoritmo `C4.5` a partir del conjunto de entrenamiento. Se calcula el porcentaje de acierto utilizando el conjunto de test asociado al conjunto de entrenamiento, empleando el modelo extraído a partir de él por el algoritmo `C4.5`.

$$ACTEST = Porcentaje\ de\ Acierto\ en\ Test \tag{1}$$

2.2. Tamaño del Modelo

El tamaño del modelo se evalúa considerando el número de reglas (n_R) que lo componen [2].

$$TAM = n_R \tag{2}$$

2.3. Número de Antecedentes Medio de las Reglas del Modelo

Como segunda medida del tamaño introduciremos el número medio de antecedentes por regla (ver (3) y (4)) [2]. Sean las reglas de la forma $Cond \rightarrow Clase$, $N_{Antecedentes}(R_i)$ representa el número de antecedentes de la regla R_i y ANT el número medio de antecedentes.

$$N_{Antecentes}(R_i) = \sharp|Cond_i| \tag{3}$$

$$ANT = \frac{1}{n_R}\sum_{i=1}^{n_R} N_{Antecedentes}(R_i) \tag{4}$$

Tanto el tamaño como el número medio de antecedentes servirán para estudiar las prestaciones a nivel de interpretabilidad del modelo.

3. El Problema de Escalado

En esta sección vamos a estudiar de qué forma afecta el tamaño del conjunto de datos sobre los algoritmos de selección de prototipos y sobre los modelos predictivos generados.

La mayoría de algoritmos de selección de prototipos tienen problemas para poder evaluar conjuntos de datos de gran tamaño. 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 selección de prototipos 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` [11], `Snn` [12], `Shrink` [13], 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.
- *Representación.* A los `AAEE`, además de los anteriores inconvenientes, habría que añadirle el derivado de la representación que emplean para codificar sus cromosomas. Cuando el tamaño del cromosoma es demasiado grande la convergencia del algoritmo se ve penalizada así como su coste computacional.

Estos desventajas provocan una considerable degradación del comportamiento y los resultados de los algoritmos. Los algoritmos de selección de prototipos evaluados directamente sobre el conjunto completo de gran tamaño son ineficaces e ineficientes.

El tamaño de los árboles de decisión generados empleando conjuntos de entrenamiento de tamaño muy grande se ve aumentando considerablemente [14]. Los árboles de decisión con tamaños grandes producen:

- *Sobreaprendizaje.* La hipótesis aprendida se adecúa demasiado a los ejemplos de entrenamiento de tal forma que sus capacidades de generacilización se ven afectadas [15].
- *La interpretabilidad desciende.* El elevado tamaño de los árboles de decisión produce como inconveniente la excesiva complejidad del modelo, que puede convertirlo en incomprensible a los expertos [16,2].

4. Selección de Conjuntos de Entrenamiento Evolutiva Estratificada

El modelo seguido para llevar a cabo la selección de datos consiste en la combinación de la estratificación del conjunto de datos con `AAEE`. Se pretende con ello el poder aplicar la selección de prototipos en conjuntos de cualquier tamaño. Con la estratificación se reduce el espacio de búsqueda, al mismo tiempo que el componente evolutivo optimiza la exploración en él.

En la Subsección 4.1 se muestra el proceso de selección de conjuntos de entrenamiento. La Subsección 4.2 describe el empleo de los `AAEE` para la selección de conjuntos de entrenamiento, destacando el esquema de representación y la función objetivo empleada. Finalmente, en la Subsección 4.3 se ofrece la propuesta evolutiva estratificada aplicada a selección de conjuntos de entrenamiento.

4.1. Selección de Conjuntos de Entrenamiento

El objetivo perseguido es obtener conjuntos de entrenamiento tales que permitan generar modelos predictivos compuestos por reglas con capacidades elevadas de predicción e interpretabilidad (ver Figura 1).

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 `C4.5` que generará el modelo a partir del conjunto `TSS` de entrada que será validado empleando `TS`.

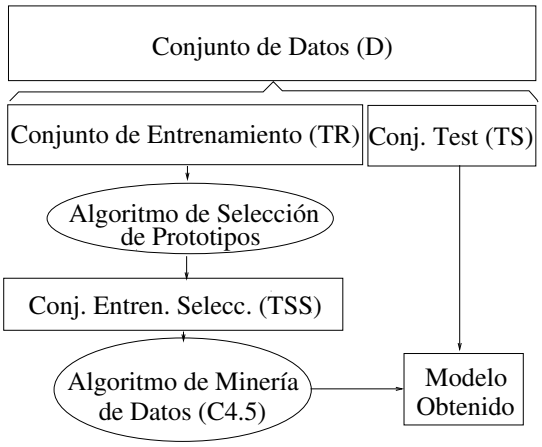

Figura 1. Selección de prototipos aplicada a selección de conjuntos de entrenamiento

4.2. Algoritmos Evolutivos Aplicados a Selección de Conjuntos de Entrenamiento

Para estudiar el empleo de `AAEE` en selección de conjuntos de entrenamiento tenemos que resaltar dos factores clave: especificar la representación y definir la función de evaluación de los cromosomas [10].

Esquema de Representación. Partimos de un conjunto `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. 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 (`porc_clas`) asociado a `TSS` y el porcentaje de reducción (`porc_red`) conseguido en `TSS` con respecto a `TR`:

$$\texttt{F\_Eval}(\texttt{TSS}) = \alpha \cdot \texttt{porc\_clas} + (1 - \alpha) \cdot \texttt{porc\_red}. \tag{5}$$

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:

$$\texttt{porc\_red} = 100 \cdot (|\texttt{TR}| - |\texttt{TSS}|)/|\texttt{TR}|. \quad (6)$$

El objetivo de los AAEE es maximizar la función de evaluación definida. Para ello deben maximizar el porcentaje de clasificación y el de reducción.

4.3. Selección Evolutiva de Conjuntos de Entrenamiento Estratificada

Tras llevar a cabo el proceso de estratificación y selección, se reúnen los subconjuntos seleccionados y se procede a aplicar el algoritmo de MDD para generar su modelo asociado. La Figura 2 muestra el proceso [17,18].

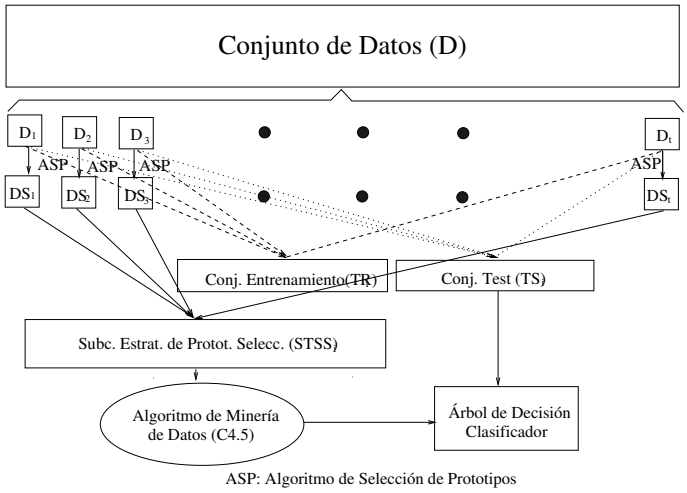

Figura 2. Selección de Prototipos Evolutiva Estratificada aplicada a Selección de Conjuntos de Entrenamiento

El conjunto de test (TS) será el complementario de TR en D. DS_j son subconjuntos seleccionados a partir de D_j (ver (7) , (8) y (9)).

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

$$TS = D \setminus TR \quad (8)$$

El conjunto STSS es la selección estratificada sobre TR. Se obtiene mediante la unión de los DS_j (ver (9)).

$$STSS = \bigcup_{j \in J} DS_j, J \subset \{1, 2, ..., t\} \quad (9)$$

La calidad en precisión del subconjunto seleccionado `STSS` se evalúa utilizándolo como entrada para el algoritmo `C4.5`, empleando como test el conjunto `TS`.

5. Estudio Experimental

Esta sección presenta el estudio experimental desarrollado para obtener modelos predictivos a partir de la selección evolutiva estratificada de conjuntos de entrenamiento.

La Subsección 5.1 muestra le metodología seguida en la experimentación. En la Subsección 5.2 se describe la estructura de la tablas que contiene los resultados, y se presenta dicha tabla. Y finálmente, en la Subsección 5.3 llevaremos a cabo el análisis de los mismos.

5.1. Metodología de Experimentación

En este apartado mostramos el conjunto de datos empleado, los algoritmos y sus correspondientes parámetros y el esquema de estratificación y particiones.

Conjunto de Datos, Algoritmos y Parámetros.

El conjunto de datos evaluado en la experimentación es Kdd Cup'99. Se trata de un conjunto de datos de gran tamaño, con objeto de destacar el problema de escalado.

La tarea que se modeliza en Kdd Cup'99 consiste en detectar 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` [19]. El conjunto se caracteriza por presentar 494022 instancias, con 41 atributos cada una, pertenecientes a 23 clases diferentes.

Dividiremos a los algoritmos evaluados en dos grupos, dependiendo de su naturaleza evolutiva. Así tenemos:

- Algoritmos no evolutivos. Los algoritmos no evolutivos empleados en este estudio son: `Cnn` [20], `Ib2` [21] e `Ib3` [21]. Han sido seleccionados por ser los más eficientes en [10]. El único que necesita fijar parámetros es el `Ib3`, dónde el nivel de *Aceptabilidad* se establece en 0.9 y el de *Eliminación* en 0.7.
- Algoritmos evolutivos. Se ha seleccionado el algoritmo `CHC` como modelo evolutivo eficaz y eficiente, basándose en el comportamiento definido en [10]. El tamaño de la población es de 50 individuos y el número de evaluaciones es 10000.

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.

La vía a seguir en la validación cruzada aparece reflejada en la Figura 2. A éste modo de validación la llamaremos validación cruzada estratificada y se denotará como Tfcv st.

En Tfcv st, cada TR_i se define, como podemos ver en (10), mediante la unión de subconjuntos D_j (ver Figura 2).

Los subconjuntos TR_i y TS_i, i=1, ..., 10 se obtienen siguiendo (10) y (11):

$$\begin{array}{c} TR_i = \bigcup_{j \in J} D_j, \\ J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \end{array} \tag{10}$$

$$TS_i = D \setminus TR_i \tag{11}$$

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.

El subconjunto $STSS_i$ se obtiene mediante la unión de conjuntos DS_j en vez de emplear D_j (ver (12)).

$$\begin{array}{c} STSS_i = \bigcup_{j \in J} DS_j, \\ J = \{j / 1 \leq j \leq b \cdot (i-1) \ y \ (i \cdot b) + 1 \leq j \leq t\} \end{array} \tag{12}$$

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

Para el conjunto Kdd Cup'99 se han empleado 100 estratos.

Como referencia, se introduce la ejecución del algoritmo C4.5 sobre el conjunto original sin reducción siguiendo un proceso de validación cruzada clásico (Tfcv cl). Así mismo, incluiremos su ejecución empleando la máxima y la mínima poda, con el objetivo de analizar la interpretabilidad de los modelos que genera con respecto a los obtenidos por el resto de algoritmos estudiados.

5.2. Resultados

En esta sección mostraremos la tabla en la que se presentan los resultados. La tabla presenta la siguiente organización:

- La primera columna muestra el nombre del algoritmo. Cada nombre aparecerá acompañado por el tipo de validación empleado, cl para la clásica y st para la estratificada. El algoritmo C4.5 se acompaña de Min y Max para indicar las evaluaciones llevadas a cabo aplicando mecanismo de poda mínima y máxima respectivamente.
- En la segunda columna encontramos el porcentaje de reducción medio conseguido por el algoritmo de selección de instancias en la validación cruzada.
- La tercera columna contiene la media del porcentaje de acierto en entrenamiento conseguido con los conjuntos de entrenamiento seleccionados.

- La cuarta columna muestra la media del porcentaje de acierto en test conseguido con los conjuntos de entrenamiento.
- La quinta columna presenta el número de reglas medio que componen los modelo.
- La sexta columna muestra el número medio de antecedentes que componen las reglas de los modelos.

Las Tabla 1 contiene los resultados asociados a la base de datos Kdd Cup'99.

Cuadro 1. Calidad de las Reglas en Kdd Cup'99. Tabla 1.

	% Reducción	%Acierto Entren.	% Acierto Test	TAM	ANT
C4.5 Min cl		99.99	99.96	252	13.34
C4.5 cl		99.97	99.95	143	11.78
C4.5 Max cl		99.95	99.95	102	10.52
Cnn st	81.61	99.92	96.43	83	11.49
Ib2 st	82.01	99.40	95.05	58	10.86
Ib3 st	78.82	98.13	96.77	74	11.48
CHC st	99.28	98.97	98.41	9	3.56

5.3. Análisis de Resultados

A continuación ofrecemos un análisis de los resultados obtenidos:

- Considerando el porcentaje de reducción conseguido con respecto al conjunto inicial, nuestra aproximación al problema de selección evolutiva de conjuntos de entrenamiento es claramente positiva, ofrece el mayor porcentaje de reducción de entre los algoritmos estudiados.
- Al mantener el conjunto de entrenamiento sin ningún tipo de reducción, el algoritmo `C4.5` es el que ofrece la mayor precisión en clasificación.
  La selección evolutiva estratificada consigue el mejor acierto de todos los algoritmos de selección.
- El tamaño del modelo, tanto en número de reglas como en número de antecedentes de las mismas, puede estar ligado al tamaño del conjunto de entrenamiento a partir del cual se genera. Cuanto mayor sea dicho conjunto, tanto por patrones como por número de variables asociadas, mayor puede ser el modelo obtenido.
  Los algoritmos de selección que ofrecen las mayores reducciones suelen conducir a la obtención de los modelos de menor tamaño como ocurre con el uso del algoritmo `CHC estratificado`, que consigue los mejores porcentajes en reducción y tiene asociados a su vez los modelos de menor tamaño. El menor tamaño de los modelos se ve reflejado en la interpretabilidad que proporcionan.

6. Conclusiones

En este capítulo se ha analizado la extracción de modelos predictivos basados en reglas a partir de la selección evolutiva de conjuntos de entrenamiento estratificada. Se ha evaluado la calidad de los modelos obtenidos considerando su precisión e interpretabilidad.

Las principales conclusiones alcanzadas son las siguientes:

- Considerando la reducción ofrecida por cada algoritmo, la selección evolutiva estratificada presenta el mejor comportamiento.
- Teniendo en cuenta el tamaño del modelo se aprecia que la versión estratificada de `CHC` ofrece los menores valores, tanto en número de reglas como en número de antecedentes. Dicha circunstancia proporciona conjuntos de reglas con mayor interpretabilidad.

Como conclusión final, consideramos que la extracción de modelos predictivos a partir la selección evolutiva estratificada de conjuntos de entrenamiento presenta buenas prestaciones. Se consiguen los modelos predictivos de menor tamaño con porcentajes de acierto elevados, cercanos a los que ofrece el algoritmo `C4.5` sin reducción. De esta forma, el algoritmo `CHC` estratificado permite obtener los modelos predictivos con mayor equilibrio interpretabilidad-precisión.

Referencias

1. Hernández, J., Ramirez, M., Ferri, C.: Introducción a la Minería de Datos. Pearson (2004)
2. Kweku-Muata, Osei-Bryson: Evaluation of decision trees: a multicriteria aproach. Cumputers and Operations Research **31** (2004) 1933–1945
3. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann (1993)
4. Wilson, D.R., Martinez, T.R.: Reduction tecniques for instance-based learning algorithms. Machine Learning **38** (2000) 257–268
5. Liu, H., Motoda, H.: On issues of instance selection. Data Mining and Knowledge Discovery **6** (2002) 115–130
6. Back, T., Fogel, D., Michalewicz, Z.: Handbook of evolutionary computation. Oxford University Press (1997)
7. Kuncheva, L.: Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognition Letters **16** (1995) 809–814
8. Shinn-Ying, H., Chia-Cheng, L., Soundy, L.: Design of an optimal nearest neighbour classifier using an intelligent genetic algorithm. Pattern Recognition Letters **23(13)** (2002) 1495–1503
9. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. Foundations of Genetic Algorithms **1** (1991) 265–283
10. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. IEEE Transaction on Evolutionary Computation **7(6)** (2003) 561–575

11. Gates, G.W.: The reduced nearest neighbour rule. IEEE Transaction on Information Theory **18(5)** (1972) 431–433
12. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: An algorithm for a selective nearest neighbour decision rule. IEEE Transaction on Information Theory **21(6)** (1975) 665–669
13. Kibbler, D., Aha, D.W.: Learning representative exemplars of concepts: An initial case of study. In: Proc. of the Fourth International Workshop on Machine Learning. (1987) 24–30
14. Zheng, Z.: Scaling up the rule generation of c4.5. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. (1998) 348–359
15. 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). (1991) 192–205
16. Zhou, Z.H., Jiang, Y.: Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. IEEE Transactions on Information Technology in Biomedicine **7** (2003) 37–42
17. Cano, J., Herrera, F., Lozano, M.: Evolutionary algorithms in stratified instance selection for data reduction in large datasets. In: Proc. of the 8th Online World Conference on Soft Computing in Industrial Applications. (2003)
18. Cano, J., Herrera, F., Lozano, M.: Estratificación y selección evolutiva de prototipos aplicadas a bases de datos de gran tamaño. In: Actas del Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. (2004) 211–220
19. Merz, C.J., Murphy, P.M.: UCI repository of machine learning databases. (1996) University of California Irvine, Department of Information and Computer Science, http://kdd.ics.uci.edu.
20. Hart, P.E.: The condensed nearest neighbour rule. IEEE Transaction on Information Theory **18(3)** (1968) 431–433
21. Aha, D., Kibbler, D., Albert, M.: Instance-based learning algorithms. Machine Learning **6** (1991)