Un Primer Estudio sobre la Utilización de Selección Evolutiva de Conjuntos de Entrenamiento en Problemas de Clasificación con Clases no Balanceadas y Árboles de Decisión

Salvador García, Alberto Fernández, Francisco Herrera

Resumen— **La clasificación en dominios no balanceados es un desafío muy reciente dentro del ámbito de aprendizaje automático. Hablamos de clasificación no balanceada cuando los datos a tratar presentan muchos ejemplos pertenecientes a una clase y pocos ejemplos de la otra clase. Además, la clase menos representativa es la que tiene mayor interés desde el punto de vista del aprendizaje. Una de las técnicas más usadas para abordar este problema consiste en el preprocesamiento previo de los datos al proceso de aprendizaje. Este preprocesamiento puede hacerse mediante técnicas de bajo-muestreo; borrando ejemplos pertenecientes principalmente a la clase mayoritaria; y técnicas de sobre-muestreo, por medio de la replicación o generación de nuevos ejemplos de la clase minoritaria.**

Esta contribución propone un nuevo procedimiento de bajo-muestreo basado en algoritmos evolutivos para llevar a cabo una selección de ejemplos del conjunto de entrenamiento y mejorar los modelos obtenidos por algoritmos de árboles de decisión. En este estudio, hemos utilizado el algoritmo C4.5, muy conocido en ámbitos de clasificación. La propuesta ha sido comparada con otras técnicas de bajo-muestreo y sobre-muestreo y los resultados nos muestran que la propuesta es muy competitiva en términos de precisión e interpretabilidad de los modelos obtenidos.

Palabras clave— **algoritmos evolutivos, clasificación no balanceada, reducción de datos, selección de conjuntos de entrenamiento, árboles de decisión.**

I. Introducción

En los últimos años, el problema de las clases no balanceadas en clasificación es uno de los desafíos emergentes dentro del área de minería de datos [1]. El problema aparece cuando los datos presentan un llamado no balanceo de clases, es decir, cuando la distribución de ejemplos entre las distintas clases es variable, siendo usual que las clases menos representativas sean las más interesantes desde el punto de vista del aprendizaje [2]. El no balanceo en la distribución de clases está presente en una alta variedad de aplicaciones del mundo real, incluyendo, pero no limitado, a telecomunicaciones, WWW, financias, biología y medicina.

Normalmente, los ejemplos se agrupan en dos tipos o clases: la clase mayoritaria o clase negativa, y la clase minoritaria o clase positiva. Como comentábamos anteriormente, la clase minoritaria o positiva es aquella que tiene mayor interés y suele estar asociada a mayores costes ocasionados en el caso de una clasificación errónea. Un clasificador convencional podría ignorar la importancia de la clase minoritaria porque su representación dentro del conjunto de datos no es tan significativa como debiera. A modo de ejemplo clásico, si la tasa de no balanceo presentado en los datos es 1:99 (esto es, hay una instancia positiva por cada 99 instancias negativas), el resultado de ignorar los ejemplos de la clase positiva repercutiría en un error sólo del 1 %.

Muchos métodos han sido propuestos para tratar con el problema de las clases no balanceadas. Éstos pueden ser divididos en técnicas algorítmicas y técnicas basadas en datos. Las primeras suponen realizar modificaciones en el funcionamiento de los algoritmos, haciéndolos sensibles al coste para beneficiar la clase minoritaria [3], [4]. Las técnicas basadas en datos modifican la distribución de datos condicionadas a una función de evaluación. El muestreo de los datos se puede hacer por medio del bajo-muestreo, borrando ejemplos de los datos, y por medio del sobre-muestreo, replicando o generando nuevos ejemplos minoritarios. Existen numerosos trabajos y casos de estudio que muestran las ventajas de cada uno de ellos [5], [6], [7], [8], [9].

Los Algoritmos Evolutivos (AEs) han sido usados para la reducción de datos en aprendizaje automático con excelentes resultados. Se han utilizado tanto para selección de características [10], [11], [12], como selección de instancias [13], [14]. También se han utilizado para realizar bajo-muestreo de los datos en dominios no balanceados con aprendizaje basado en instancias [15]. Los AEs obtienen un buen comportamiento para la Selección de Conjuntos de Entrenamiento (SCE) en términos de obtener un buen equilibrio entre precisión e interpretabilidad con reglas de clasificación [16].

En esta contribución, proponemos el uso de AEs para SCE en clasificación con conjuntos de datos no balanceados. Nuestro objetivo es incrementar la tasa

Universidad de Granada. Departamento de Ciencias de la Computación e Inteligencia Artificial 18071 Granada, España. E-mail: {salvagl,alberto,herrera}@decsai.ugr.es.

de acierto de los algoritmos basados en árboles de decisión. Hemos utilizado el algoritmo más comúnmente empleado en la práctica, C4.5 [17]. Para incrementar su eficacia, se procederá a borrar ejemplos del conjunto de entrenamiento que principalmente pertenezcan a la clase mayoritaria. Comparamos nuestra propuesta con otras técnicas de bajo-muestreo, sobre-muestreo e hibridaciones más avanzadas entre bajo-muestreo y sobre-muestreo estudiadas en la literatura especializada [5]. El estudio experimental ha sido contrastado a través de técnicas estadísticas no paramétricas.

Para alcanzar dicho objetivo, el resto de esta contribución se organiza de la siguiente manera: La Sección II da una explicación sobre la medida utilizada para evaluar clasificadores en problemas no balanceados. En la Sección III, los conceptos de la SCE evolutiva son explicados, junto con la descripción del modelo utilizado. En la Sección IV, se describe el marco experimental y se presentan los resultados y su análisis. Finalmente, en la Sección V, señalamos nuestras conclusiones del trabajo.

II. Medida de Evaluación Empleada para Analizar los Problemas con Clases no Balanceadas

Cuando queremos evaluar un clasificador sobre dominios de aprendizaje no balanceados, la formas clásicas de evaluación, como por ejemplo la tasa de acierto de clasificación, no tienen sentido. Un clasificador convencional que usa la tasa de acierto podría mostrar una tendencia favorable a la clase mayoritaria debido al comportamiento inherente que hay en la propia medida, el cual está directamente relacionado con la tasa entre el número de instancias de cada clase.

La forma más correcta de evaluar el rendimiento de los clasificadores en este dominio está basada en el análisis de la matriz de confusión. En la Tabla I, se ilustra una matriz de confusión para un problema de dos clases con los valores para las clases positiva y negativa. Desde esta matriz, es posible extraer un gran número de métricas para medir el rendimiento de un sistema de aprendizaje, como la *Tasa de Error*, definida como $Err = \frac{FP+FN}{VP+FN+FP+VN}$ y la *Tasa de Acierto*, definida como $Acc = \frac{VP+VN}{VP+FN+FP+VN} = 1 - Err$

TABLA I
Matriz de Confusión para un Problema de dos Clases

	Predicción Positiva	*Predicción Negativa*
Clase Positiva	Verdadero Positivo (VP)	Falso Negativo (FN)
Clase Negativa	Falso Positivo (FP)	Verdadero Negativo (VN)

En los problemas no balanceados, es más correcto tratar la precisión de las clases de un modo independiente. Así, a partir de la Tabla I, se pueden extraer las siguientes medidas:

- **Tasa de falsos negativos** $FN_{tasa} = \frac{FN}{VP+FN}$ es el porcentaje de casos verdaderos positivos mal clasificados como negativos.
- **Tasa de falsos positivos** $FP_{tasa} = \frac{FN}{FP+VN}$ es el porcentaje de casos verdaderos negativos mal clasificados como positivos.
- **Tasa de verdaderos negativos** $TN_{tasa} = \frac{VN}{FP+VN}$ es el porcentaje de casos verdaderos negativos correctamente clasificados como negativos.
- **Tasa de verdaderos positivos** $TP_{tasa} = \frac{VP}{VP+FN}$ es el porcentaje de cases verdaderos positivos correctamente clasificados como positivos.

El objetivo de un clasificador es minimizar las tasa de falsos positivos y falsos negativos o, de una forma análoga, maximizar la tasa de verdaderos positivos y verdaderos negativos.

En [18] se utiliza una métrica llamada *Media Geométrica (MG)* de las tasas individuales de acierto, definida como $g = \sqrt{a^+ \cdot a^-}$, donde a^+ denota la tasa de acierto en los ejemplos positivos (VP_{tasa}), y a^- es la tasa de acierto sobre los ejemplos negativos (VN_{tasa}). Se trata de una medida de evaluación que permite simultáneamente maximizar la tasa de acierto en ejemplos positivos y negativos con un buen equilibrio entre ambos. Nuestro estudio estará centrado por tanto en esta medida.

III. Selección Evolutiva de Conjuntos de Entrenamiento en Clasificación con Clases no Balanceadas

Asumamos que existe un conjunto de entrenamiento TR con N instancias que consisten en parejas $(x_i, y_i), i = 1, ..., N$, donde x_i define un vector de entrada de atributos e y_i define la correspondiente etiqueta de clase. Cada una de las N instancias tiene M atributos de entrada y deben pertenecer a la clase positiva o negativa. Definimos $S \subseteq TR$ como un subconjunto de instancias seleccionadas resultado de la ejecución de un algoritmo de muestreo.

La SCE puede considerarse como un problema de búsqueda en el que los AEs pueden ser aplicados. Nuestra propuesta será denominada por Bajo-Muestreo Evolutivo para Selección de Conjuntos de Entrenamiento (BMESCE). Tenemos en cuenta dos conceptos importantes: la especificación de las soluciones y la definición de la función objetivo.

- *Representación:* El espacio de búsqueda asociado se constituye por todos los subconjuntos de TR. Esto se consigue usando una representación binaria. Un cromosoma contiene N genes (uno por cada instancia en TR) con dos posibles estados: 0 y 1. Si el gen vale 1, su instancia asociada se incluye en el subconjunto de TR representado por el cromosoma. Si vale 0, esto no sucede.
- *Función Objetivo:* Sea S el subconjunto de instancias de TR que está codificado por un cromosoma. Definimos una función objetivo basada en la medida MG y evaluada sobre TR.

$$Objetivo(S) = MG. \quad (1)$$

Esta función objetivo se relaciona con la propuesta denominada Bajo-Muestreo Evolutivo guiado por

Medidas de Clasificación (BMEMC), propuesto en [15]. El árbol de decisión C4.5 se utiliza para medir la tasa de acierto asociada con el árbol inducido usando las instancias seleccionadas en S. La tasa de acierto independientemente calculada en cada clase es útil para obtener el valor MG asociado con un cromosoma. El objetivo de los AEs para este problema consiste en maximizar la función objetivo definida: maximizar la tasa MG.

Es necesario incluir un mecanismo dentro de la función objetivo para evitar el posible sobreaprendizaje. Aunque C4.5, en su definición estándar, incorpora un mecanismo de poda para evitar sobreajuste, la integración del proceso de inducción de un árbol dentre del ciclo evolutivo puede dirigir el árbol resultante a ser un modelo óptimo para datos de entrenamiento, perdiendo la capacidad de generalización sobre datos de test. Incorporamos un mecanismo simple y efectivo que consiste en proporcionar a los costes de clasificación un mayor peso (W) a las instancias que no están incluidas en S que a las instancias que están incluidas en S. Una instancia de TR bien clasificada suma un valor W si no está incluida en S, y un valor de 1 si lo está. El procedimiento apoya la capacidad de reducción del subconjunto seleccionado, dado que es más beneficioso evaluar cromosomas con un mayor número de ejemplos fuera que dentro del subconjunto seleccionado. Obviamente, la instancia produce una penalización en tasa de acierto de la misma magnitud en caso de estar mal clasificada. Nuestros estudios empíricos han determinado que un valor de W igual a 3 funciona de una manera adecuada.

▪ *Operador de Cruce para Reducción de Datos:* Para alcanzar una buena tasa de reducción, el Cruce Heurístico Uniforme (HUX: Heuristic Uniform Crossover) implementado por CHC sufre un cambio que hace más difícil la inclusión de instancias dentro del subconjunto seleccionado. Por tanto, si un cruce HUX pone un bit a 1 en un gen, el bit puede volver a valer 0 dependiendo de una determinada probabilidad (su valor será especificado en la Sección IV-A, Tabla III).

▪ Como método de computación evolutiva, hemos usado el modelo CHC [19], [16]. CHC es un algoritmo evolutivo clásico que introduce diferentes características para obtener un buen equilibrio entre exploración y explotación del espacio de búsqueda; tales como la prevención de incesto, reinicialización del proceso de búsqueda cuando se estanca y la competición entre padres e hijos dentro del proceso de reemplazamiento.

Durante cada generación, CHC realiza los siguientes pasos.

- Usa una población de padres de tamaño N para generar una población intermedia de N individuos, los cuales son aleatoriamente emparejados y usados para generar N potenciales hijos.
- A continuación, se realiza una competición de supervivencia donde los mejores N cromosomas entre la población de padres e hijos son seleccionados para formar la siguiente generación.

CHC también implementa una forma de recombinación heterogénea usando HUX, un operador de cruce especial. HUX intercambia la mitad de los bits que difieren entre los padres, donde la posición del bit a intercambiar es aleatoriamente escogida. CHC también emplea un método de prevención de incesto. Antes de aplicar HUX a dos padres, se calcula la distancia de Hamming entre ellos. Sólo aquellos padres que difieren del otro en un determinado número de bits (umbral de emparejamiento) son recombinados. El valor inicial del mencionado umbral se establece en $L/4$, donde L es la longitud de los cromosomas. Si ningún hijo se ha insertado en la nueva población, el umbral es reducido en uno.

No se aplica mutación durante la etapa de recombinación. En vez de esto, cuando la población converge o la búsqueda no progresa (ej., el umbral de emparejamiento vale cero y ningún nuevo hijo ha sido generado siendo mejor que cualquier miembro de la población de padres) la población es reinicializada para introducir nueva diversidad en la búsqueda. El cromosoma que representa la mejor solución encontrada en el curso de la búsqueda se usa como plantilla para generar la nueva población. Este proceso se lleva a cabo cambiando aleatoriamente el 35 % de los bits del cromosoma plantilla para formar cada uno de los $N-1$ cromosomas restantes. La búsqueda es reanudada a continuación.

IV. Marco Experimental y Resultados

Esta sección describe la metodología seguida en el estudio experimental de las técnicas de remuestreo comparadas. Explicaremos la configuración del experimento: conjuntos de datos utilizados y parámetros para los algoritmos. Los algoritmos que participan en el estudio comparativo son: OSS [20], NCL [21], SMOTE [6], SMOTE + Tomek Links (TL) y SMOTE + ENN [5].

A. Marco Experimental

El rendimiento de los algoritmos se analiza utilizando 25 conjuntos de datos tomados del Repositorio de Conjuntos de Datos para Aprendizaje Automático UCI [22]. Los conjuntos de datos multi-clase se modifican para obtener problemas no balanceados de dos clases, definiendo una o la union de dos o más clases como la clase positiva y una o más clases como la clase negativa.

Las principales características de los data sets utilizados se muestran en la Tabla II. Para cada conjunto de datos, se muestra el número de ejemplos (#Ejemplos), número de atributos (#Atributos) y nombre de cada clase (minoritaria y mayoritaria).

Los conjuntos de datos considerados se particionan usando el procedimiento *ten fold cross-validation (10-fcv)*. Los parámetros de los algoritmos

TABLA II
CONJUNTOS DE DATOS NO BALANCEADOS

Conjunto de Datos	#Ejemplos	#Atributos	Clase (min., may.)	%Clase(min.,may.)
Abalone9-18	731	9	(18, 9)	(5.75, 94.25)
Dermatology2	366	34	(2, remainder)	(16.67, 83.33)
EcoliCP-IM	220	7	(im, cp)	(35.00, 65.00)
EcoliIM	336	7	(im, remainder)	(22.92, 77.08)
EcoliIMU	336	7	(iMU, remainder)	(10.42, 89.58)
EcoliOM	336	7	(om, remainder)	(6.74, 93.26)
German	1000	20	(1, 0)	(30.00, 70.00)
GlassBWFP	214	9	(build-window-float-proc, remainder)	(32.71, 67.29)
GlassBWNFP	214	9	(build-window-non_float-proc, remainder)	(35.51, 64.49)
GlassNW	214	9	(non-windows glass, remainder)	(23.93, 76.17)
GlassVWFP	214	9	(Ve-win-float-proc, remainder)	(7.94, 92.06)
Haberman	306	3	(Die, Survive)	(26.47, 73.53)
New-thyroid	215	5	(hypo,remainder)	(16.28, 83.72)
PageBlocks(2,4,5)-3	559	10	(3, 2+4+5)	(5.01, 94.99)
Pima	768	8	(1,0)	(34.77, 66.23)
Segment1	2310	19	(1, remainder)	(14.29, 85.71)
VehicleVAN	846	18	(van, remainder)	(23.52, 76.48)
Vowel0	990	13	(0, remainder)	(9.01, 90.99)
Yeast(1)	467	8	(POX, MIT+ME3+EXC+ERL)	(4.28, 95.72)
Yeast(2)	1240	8	(POX+ERL, MIT+NUC+CYT+ME1+EXC)	(2.02, 97.98)
Yeast(3)	1334	8	(EXC, MIT+NUC+CYT+ME3)	(2.62, 97.38)
Yeast(4)	1120	8	(VAC, NUC+CYT+ME3+EXC)	(2.68, 97.32)
YeastCYT-POX	483	8	(POX, CYT)	(4.14, 95.86)
YeastNUC-POX	449	8	(POX, NUC)	(4.45, 95.55)
YeastPOX	1484	8	(POX, remainder)	(1.35, 98.65)

se muestran en la Tabla III.

TABLA III
PARÁMETROS CONSIDERADOS PARA LOS ALGORITMOS.

Algoritmo	Parámetros
SMOTE	$k = 5,\ Tasa\ Balance = 1:1$
BMESCE	$Pob = 50,\ Eval = 10000,$ $Prob.\ inclusion\ HUX = 0{,}25,\ W = 3$

B. Resultados y Análisis

La Tabla IV muestra los resultados en datos de entrenamiento y la Tabla V muestra los resultados en datos de test obtenidos por los algoritmos comparados utilizando la medida de evaluacion MG. En ambos casos, la columna denominada *ninguno* se corresponde con el caso en el que ningún método de remuestreo se ha ejecutado previo a C4.5. El mejor caso en cada conjunto de dato está destacado en negrita.

La Tabla VI muestra el número medio de reglas (u hojas) obtenido por C4.5 en cada uno de los conjuntos de datos.

Observando las Tablas IV, V y VI, podemos hacer el siguiente análisis:

- En entrenamiento, los resultados se inclinan hacia el algoritmo SMOTE y SMOTE+ENN, principalmente. Sin embargo, cuando observamos los resultados obtenidos en test, vemos como SMOTE, en media, pierde rendimiento frente a las técnicas híbridas o BMESCE. Esto indica que, aunque existe sobreaprendizaje en todas las técnicas, con SMOTE es más notable.
- La propuesta BMESCE obtiene el mejor resultado de test en media considerando la medida MG. Claramente mejora los otros métodos de bajo-muestreo (OSS y NCL) y mejora la tasa de acierto incluso al compararlo con las técnicas de sobre-muestreo.
- Las técnicas de sobre-muestreo obtienen una mejor tasa de acierto que los procedimiento de bajo-muestreo con C4.5 [5], pero no alcanzan a mejorar la propuesta BMESCE.
- Excepto para NCL, BMESCE produce árboles de decisión con un menor número de hojas que los restantes métodos. Por otro lado, aunque la combinación NCL + C4.5 obtiene árboles más pequeños, la tasa de MG es la más baja de todos los métodos de remuestreo comparados.
- Las técnicas de sobremuestreo obligan a C4.5 a producir árboles con más hojas. Este hecho no es deseable cuando nuestro interés está en obtener modelos interpretables.

Hemos incluido un segundo tipo de tabla que ofrece una comparación estadística de métodos sobre múltiples conjuntos de datos. Demšar [23] recomienda un conjunto simple, seguro y robusto de tests no paramétricos para hacer comparaciones estadísticas entre clasificadores. Uno de ellos es el Test de Rangos y Signos de Wilcoxon [24]. La Tabla VII ofrece los resultados de aplicar el test de Wilcoxon entre nuestra propuesta y cada uno de los restantes algoritmos de remuestreo estudiados en este trabajo sobre los 25 conjuntos de datos considerados. Esta tabla se divide en dos partes: En la primera la medida de rendimiento usada es la tasa de acierto a través de la medida MG. En la segunda parte, aplicamos el test de Wilcoxon usando como medida de rendimiento el número de reglas u hojas producidas por C4.5. Cada parte de esta tabla contiene una columna, que representa nuestra propuesta, y N_A filas donde N_A es el número de algoritmos considerados en el estudio. En cada una de las celdas puede aparecer tres símbolos: +, = ó -. Representan que la propuesta mejora (+), es similar (=) o es peor (-) en rendimiento que el algoritmo que aparece en la primera columna (Tabla VII). Los valores entre paréntesis es el valor p obtenido en la comparación y el nivel de significancia considerado es $\alpha = 0{,}05$.

A continuación, hacemos un breve análisis de los

TABLA IV
RESULTADOS OBTENIDOS POR C4.5 USANDO MG SOBRE DATOS DE ENTRENAMIENTO

datos	ninguno	NCL	OSS	SMOTE	SMOTE + ENN	SMOTE + TL	BMESCE
abalone9-18	0.6611	0.7206	0.7218	**0.9348**	0.9337	0.8543	0.8449
dermatology2	0.9563	0.9240	0.9437	**0.9894**	0.9853	0.9845	0.9820
ecoliCP-IM	0.9869	0.9526	0.9869	**0.9906**	0.9860	0.9862	0.9869
ecoliIM	0.8602	0.9184	0.9275	**0.9502**	0.9483	0.9341	0.9428
ecoliMU	0.8794	0.8799	0.9234	**0.9722**	0.9625	0.9331	0.9374
ecoliOM	0.9416	0.9197	0.9576	0.9782	0.9891	0.9566	**0.9914**
german	0.7779	0.6881	0.7790	**0.8676**	0.8136	0.7773	0.7474
glassBWFP	0.9391	0.7557	0.8528	**0.9553**	0.8915	0.8906	0.9157
glassBWNFP	0.8684	0.6501	0.8766	**0.9450**	0.8964	0.8720	0.8856
glassNW	0.9770	0.8456	0.9670	**0.9899**	0.9679	0.9704	0.9783
glassVWFP	0.8476	0.8828	0.9691	**0.9779**	0.9611	0.8968	0.9608
haberman	0.4660	0.4856	0.7215	**0.7733**	0.7519	0.7520	0.7141
new-thyroid	0.9678	0.9507	0.9787	0.9869	0.9873	0.9854	**0.9963**
pageblocks(2,4,5)-3	0.9919	0.9542	0.9918	**1.0000**	**1.0000**	0.9980	**1.0000**
pima	0.8151	0.7115	0.8115	**0.8631**	0.8387	0.8210	0.8084
segment1	0.9908	0.9827	0.9957	**0.9991**	0.9988	0.9972	0.9969
vehicle	0.9856	0.8965	0.9696	**0.9889**	0.9784	0.9713	0.9666
vowel0	0.9973	0.9531	0.9973	0.9941	0.9949	0.9947	**0.9979**
yeast(1)	0.6699	0.7491	0.6171	**0.9467**	0.9460	0.8769	0.9357
yeast(2)	0.3938	0.7902	0.4203	0.8888	0.8918	0.8668	**0.8936**
yeast(3)	0.8862	0.9053	0.8973	**0.9642**	0.9675	0.9334	0.9554
yeast(4)	0.1086	0.1460	0.4341	0.7927	**0.8241**	0.6912	0.7793
yeastCYT-POX	0.2568	0.8052	0.3438	0.9072	0.9205	0.8793	**0.9377**
yeastNUC-POX	0.6742	0.8265	0.6742	0.9215	0.9379	0.8970	**0.9745**
yeastPOX	0.0000	0.7362	0.0000	0.8279	**0.8502**	0.8220	0.8473
MEDIA	*0.7560*	*0.8012*	*0.7903*	**0.9362**	*0.9289*	*0.9017*	*0.9191*

TABLA V
RESULTADOS OBTENIDOS POR C4.5 USANDO MG SOBRE DATOS DE TEST

datos	ninguno	NCL	OSS	SMOTE	SMOTE + ENN	SMOTE + TL	BMESCE
abalone9-18	0.3763	0.4761	0.4963	0.6023	**0.6724**	**0.6724**	0.6697
dermatology2	0.8623	0.8988	0.8928	0.9194	0.9181	0.9098	**0.9505**
ecoliCP-IM	**0.9787**	0.9486	**0.9787**	0.9751	0.9748	**0.9787**	**0.9787**
ecoliIM	0.8167	0.8882	0.8860	0.8795	**0.9060**	0.8811	0.8809
ecoliMU	0.7709	0.7600	0.8092	0.8661	0.8137	**0.8671**	0.8579
ecoliOM	0.8073	0.8220	0.8749	0.8412	0.8010	0.8725	**0.9291**
german	0.5759	0.6437	**0.6753**	0.6410	0.6636	0.6658	0.6419
glassBWFP	0.8138	0.6652	0.7551	0.8216	0.7599	0.7971	**0.8425**
glassBWNFP	0.6934	0.5648	0.7353	0.7511	**0.7631**	0.7427	0.7235
glassNW	0.8942	0.8101	**0.9505**	0.9239	0.9373	0.9344	0.9321
glassVWFP	0.5286	0.6755	0.6884	0.6994	0.7572	0.4930	**0.7816**
haberman	0.4280	0.4329	0.6089	**0.6832**	0.6292	0.6022	0.6206
new-thyroid	0.9048	0.9132	0.8810	0.9193	**0.9492**	0.9414	0.9463
pageblocks(2,4,5)-3	0.9270	0.9327	0.9260	**0.9991**	**0.9991**	0.9807	**0.9991**
pima	0.6908	0.6457	0.7161	0.7155	0.6990	**0.7181**	0.7179
segment1	0.9852	0.9728	0.9849	0.9918	0.9947	0.9965	0.9891
vehicle	0.9172	0.8737	0.9118	0.9202	0.9216	**0.9241**	0.9239
vowel0	**0.9808**	0.9360	**0.9808**	0.9657	0.9764	0.9671	0.9734
yeast(1)	0.4121	0.5979	0.3414	0.5399	0.6073	**0.6883**	0.6271
yeast(2)	0.1155	0.7038	0.2151	0.6783	0.6940	**0.7477**	0.6846
yeast(3)	0.7343	0.8653	0.8313	0.7983	**0.8890**	0.8649	0.8759
yeast(4)	0.0000	0.0000	0.1144	0.3737	**0.4509**	0.3044	0.3749
yeastCYT-POX	0.0699	**0.7245**	0.1000	0.5585	0.6156	0.6176	0.6489
yeastNUC-POX	0.5828	0.6151	0.5536	**0.6974**	0.6630	0.5647	0.6819
yeastPOX	0.0000	0.6238	0.0000	0.5718	0.5408	**0.6410**	0.6154
MEDIA	*0.6347*	*0.7196*	*0.6763*	*0.7733*	*0.7839*	*0.7749*	**0.7947**

TABLA VII
RESULTADOS DEL TEST DE WILCOXON SOBRE MG Y NÚMERO DE REGLAS

algoritmo	BMESCE MG	BMESCE núm. reglas
none	+ (.000)	= (.447)
OSS	+ (.001)	= (.316)
NCL	+ (.000)	- (.001)
SMOTE	+ (.011)	+ (.000)
SMOTE + TL	= (.317)	+ (.000)
SMOTE + ENN	= (.391)	+ (.000)

resultados obtenidos en la Tabla VII:

- El uso del test de Wilcoxon confirma la mejora ocasionada por BMESCE sobre OSS y NCL en métodos de bajo-muestreo. También mejora a SMOTE, pero no a los híbridos derivados de dicha técnica. Hemos visto como en la Tabla V, SMOTE obtiene en media unos resultados muy similares a SMOTE + TL. Aún así, el test de Wilcoxon nos indica que tiene un comportamiento irregular dependiendo de los conjuntos de datos usados.
- En el caso de la interpretabilidad, el test de Wilcoxon de nuevo confirma los resultados observados en la Tabla VI. La combinación BMESCE + C4.5

TABLA VI
NÚMERO MEDIO DE REGLAS (HOJAS) OBTENIDO POR EL ÁRBOL DE DECISIÓN C4.5

datos	ninguno	NCL	OSS	SMOTE	SMOTE + ENN	SMOTE + TL	BMESCE
abalone9-18	8.10	6.50	7.30	57.50	57.30	52.60	6.30
dermatology2	10.6	5.4	8.9	15.5	14.3	14.5	7.2
ecoliCP-IM	2.00	2.50	2.00	2.90	3.10	2.00	2.00
ecoliIM	5.30	5.10	6.20	10.40	10.10	10.40	6.00
ecoliMU	10.00	5.80	6.50	16.70	13.10	14.00	5.40
ecoliOM	3.90	3.40	4.40	7.80	6.60	6.80	5.40
german	91.00	35.30	57.60	159.90	121.00	82.40	33.60
glassBWFP	12.20	5.80	6.70	15.70	10.40	10.40	7.00
glassBWNFP	12.40	5.50	11.60	19.90	15.90	15.90	9.60
glassNW	6.70	4.10	4.40	9.70	6.90	7.10	5.60
glassVWFP	7.50	6.10	8.40	13.40	13.10	13.50	6.90
haberman	2.60	3.90	8.70	16.10	18.20	18.00	5.70
new-thyroid	4.10	2.60	4.30	4.90	4.90	5.00	4.30
pageblocks(2,4,5)-3	4.7	3.1	4.7	4.2	4.2	4.2	4
pima	22.40	16.10	24.60	39.50	38.90	34.90	14.50
segment1	10	8.9	12.4	12.5	12.3	12.6	7.5
vehicle	20.60	12.50	16.30	28.40	23.40	22.50	11.10
vowel0	7.80	5.00	7.80	10.70	11.40	10.50	7.90
yeast(1)	3	2.2	3.2	21.2	21.9	19.2	8.2
yeast(2)	3	3.9	3.1	38.9	39	36.7	7
yeast(3)	5	4.2	3.3	32.6	29.5	28.8	5
yeast(4)	1.4	1.3	5	58.2	61.7	54.2	7.4
yeastCYT-POX	1.70	3.70	2.30	23.30	19.70	21.20	7.60
yeastNUC-POX	2.9	4.2	3	15.1	15.9	18.5	8
yeastPOX	0	2	0	34.7	36.2	36.8	5
MEDIA	*10.36*	**6.36**	*8.91*	*26.79*	*24.36*	*22.11*	*7.93*

produce un menor número de reglas u hojas en los árboles de decisión.

- BMESCE mejora OSS, NCL y SMOTE considerando la medida MG, y se comporta de forma similar a SMOTE + TL y SMOTE + ENN. Sin embargo, el número de hojas de los árboles C4.5 producidos cuando se aplica después de BMESCE es mucho menor que los producidos por la técnicas de sobre-muestreo híbridas. BMESCE permite indicur árboles muy precisos con pocas reglas.

V. Conclusiones

El propósito de este trabajo consiste en presentar una propuesta de Selección Evolutiva de Conjuntos de Entrenamiento para Árboles de Decisión en problemas de clasificación con clases no balanceadas. La propuesta, aplicada al algoritmo C4.5, nos permite obtener árboles muy precisos en este tipo de problemas con un bajo número de reglas u hojas. La precisión del modelo obtenido es muy competitiva en comparación a los métodos más avanzados de sobre-muestreo híbridos y bajo-muestreo, y la interpretabilidad de los modelos obtenidos se incrementa significativamente.

Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Educación Español, en el marco del proyecto TIN2005-08386-C05-01.

Referencias

[1] Q. Yang and X. Wu, "10 challenging problems in data mining research.," *International Journal of Information Technology & Decision Making*, vol. 5, no. 4, pp. 597–604, 2006.

[2] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets.," *SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.

[3] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule-based systems for imbalanced data sets.," *Soft Computing. In press. DOI: 10.1007/s00500-008-0319-7*, 2008.

[4] K. Huang, H. Yang, I. King, and M. R. Lyu, "Imbalanced learning with a biased minimax probability machine.," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 36, no. 4, pp. 913–923, 2006.

[5] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations*, vol. 6, no. 1, pp. 20–29, 2004.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique.," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[7] G. M. Weiss and F. J. Provost, "Learning when training data are costly: The effect of class distribution on tree induction.," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[8] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[9] Nitesh V. Chawla, David A. Cieslak, Lawrence O. Hall, and Ajay Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Mining and Knowledge Discovery*, vol. 17, pp. 225–252.

[10] D. Whitley, R. Beveridge, C. Guerra, and C. Graves, "Messy genetic algorithms for subset feature selection.," in *Proceedings of the International Conference on Genetic Algorithms*, 1998, pp. 568–575.

[11] C. Guerra-Salcedo, Stephen Chen, D. Whitley, and S. Smith, "Fast and accurate feature selection using hybrid genetic strategies.," in *CEC*, 1999, pp. 177–184.

[12] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization.," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.

[13] J. R. Cano, F. Herrera, and M. Lozano, "Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study.," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 6, pp. 561–575, 2003.

[14] S. García, J. R. Cano, and F. Herrera, "A memetic algorithm for evolutionary prototype selection: A scaling

up approach.," *Pattern Recognition*, vol. 41, no. 8, pp. 2693–2709, 2008.

[15] S. García and F. Herrera, "Evolutionary under-sampling for classification with imbalanced data sets: Proposals and taxonomy.," *Evolutionary Computation. In press.*, 2008.

[16] 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*, vol. 60, pp. 90–108, 2007.

[17] J. R. Quinlan, *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*, Morgan Kaufmann, 1993.

[18] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems.," *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[19] L. J. Eshelman, "The CHC adaptive search algorithm: How to safe search when engaging in nontraditional genetic recombination.," in *Foundations of genetic algorithms*, G. J. E. Rawlings, Ed., pp. 265–283. 1991.

[20] M. Kubat and S. Matwin, "Addressing the course of imbalanced training sets: One-sided selection.," in *ICML*, 1997, pp. 179–186.

[21] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *AIME '01: Proceedings of the 8th Conference on AI in Medicine in Europe*, 2001, pp. 63–66.

[22] A. Asuncion and D.J. Newman, "UCI machine learning repository," 2007.

[23] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[24] D. Sheskin, *Handbook of parametric and nonparametric statistical procedures.*, Chapman & Hall/CRC, 2006.