Intercambio de Clases de acuerdo a la Distancia al Enemigo más Cercano para Problemas con Clases altamente No Balanceadas

Sergio González[1], Marcelino Lázaro[2], Anibal R. Figueiras-Vidal[2], Salvador García[1], and Francisco Herrera[1]

[1] Dept. de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada. 18071, Granada, España
[2] Dept. de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid. 28903 Getafe, Madrid
{sergiogvz,salvagl,herrera}@decsai.ugr.es,{mlazaro,arfv}@tsc.uc3m.es

Resumen La clasificación de datos no balanceados ha sido ampliamente estudiada durante años por profesionales del aprendizaje automático, considerándose uno de los retos más importantes de este campo. Los métodos basados en clasificadores o *ensembles* han resultado de gran valía para este problema. El Intercambio de Clases o *Class Switching* ha demostrado, por si solo, ser de utilidad con conjuntos de datos poco desequilibrados. En esta contribución, se ha analizado su potencial sobre problemas altamente no balanceados. Se ha desarrollado una nueva propuesta basada en *Switching* en la que se consigue el equilibrio entre clases mediante el intercambio de la salida de algunas instancias. Esto es posible gracias al novedoso método de selección basado en la Distancia al Enemigo más Cercano o *Nearest Enemy Distance (NED)*. Usando SwitchingNED como ensemble base, se han evaluado tres posibles combinaciones con técnicas de muestreo. Se ha comparado SwitchingNED con la propuesta más relevante dentro del estado del arte, resultando ligeramente mejor, a la vez que más eficiente.

Keywords: Clasificación no balanceada, Ensembles, pre-procesamiento, Class Switching.

1. Introducción

La clasificación de conjuntos de datos con una distribución de clases desequilibrada ha atraído la atención de los profesionales de la minería de datos durante años. El problema de clases no balanceadas se refiere a aquellos conjuntos con una gran disparidad en el número de instancias de cada clase. En escenarios de desequilibrio binarios, una de las clases se considera clase minoritaria, cuando está representada con muchas menos instancias que la considerada como clase mayoritaria. Dicha diferencia causa una gran perdida en el acierto de la clase minoritaria por las características de generalización de los clasificadores estándares.

En aplicaciones reales, la clase minoritaria suele ser la clase objetivo, suponiendo un gran coste su predicción incorrecta.

Por ello, se ha desarrollado una gran número de técnicas para lidiar con este problema. Se pueden clasificar en tres grupos: adaptación de clasificadores estándares al problema [12], aprendizaje sensible al coste [7] y preprocesamiento por muestreo de conjunto de datos [10]. Las técnicas basadas en conjuntos de clasificadores o ensembles se han combinado de manera exitosa con los métodos anteriormente mencionados, obteniendo los mejores resultados hasta el momento [4]. EUSBoost se presentó en [5] como el mejor ensemble para conjuntos de datos altamente desequilibrados.

En [3], se propuso un nuevo ensemble, conocido como Output Flipping, basado en la distribución al azar de las salidas del conjunto de entrenamiento, manteniendo los atributos de entrada intactos. Output Flipping intercambia las clases de las instancias, preservando su distribución original. Esta última característica limita su aplicación a conjuntos de datos no balanceados. Posteriormente, dicho concepto fue extendido en [9], recibiendo el nombre de Class Switching. Dicho algoritmo selecciona aleatoriamente las instancias que van a ser cambiadas de clase, sin mantener la distribución de clases. Estas propuestas se comportan de manera similar al Bagging [2]. Sin embargo, las particularidades de Switching tienden a encontrar el equilibrio en la distribución de las clases, rindiendo mejor que otros ensembles en escenarios poco balanceados.

Gracias a ese potencial inicial de Switching, se ha decidido, para esta contribución, explorar su idoneidad para el problema de clasificación no balanceada, especialmente en escenarios altamente desequilibrados. Se ha propuesto un nuevo ensemble partiendo de Switching con una técnica novedosa de selección de las instancias intercambiadas basadas en la Distancia al Enemigo más Cercano o Nearest Enemy Distance (NED). Esta propuesta recibe el nombre de SwitchingNED. Su nueva manera de seleccionar las instancias cambia drásticamente la idea inicial de los algoritmos de intercambio de clases, resolviendo sus desventajas hacia el problema de desequilibrio. El intercambio solo se realiza a ejemplos de la clase mayoritaria y de acuerdo a su proximidad a la clase minoritaria, medido por la NED. Esto permite un crecimiento de la clase minoritaria cerca de las fronteras, encontrando un mejor equilibrio en la distribución de clases. Como se ha venido haciendo con los ensembles [4], se ha combinado SwitchingNED con diferentes métodos de muestreo, recabándose resultados muy prometedores.

Esta contribución está estructurada de la siguiente manera. La Sección 2 presenta las propuestas basadas en intercambio de clases, concretamente el algoritmo Switching. En la Sección 3, se describe la propuesta SwitchingNED con la técnica de selección de los ejemplos a ser cambiados basado en NED. La Sección 4 proporciona información sobre la experimentación realizada, seguido de los diferentes estudios empíricos, discutiendo y analizando sus resultados. Por último, la Sección 5 concluye el trabajo.

2. Aleatorización de salidas y Intercambio de clases

En [3], Breiman lanzó una pregunta intrigante sobre la posibilidad de obtener un rendimiento comparable a otros ensembles con solo alterar las salidas del problema, las etiquetas de clase. Su método, Output Flipping [3], demostró en [9] estar bastante limitado para conjuntos de datos no balanceados, por tener restringido su ratio de cambio a la proporción de la clase minoritaria. Por este motivo, se ha descartado esta propuesta para este estudio. Posteriormente, un nuevo ensemble basado en aleatorización de clases se propuso en [9], resolviendo dicha restricción. El algoritmo Switching difiere del Output Flipping en el mecanismo usado en el cambio de las etiquetas de salida. En cada iteración del ensemble, una fracción fija *fr* de las instancias es seleccionada al azar y cambiada a otra clase, elegida aleatoriamente. Las probabilidades de cambio quedan representadas en la siguiente matriz de transición:

$$\begin{aligned} P_{j\leftarrow i} &= fr/(K-1) \quad \text{for} \quad i \neq k \\ P_{i\leftarrow i} &= 1 - fr \end{aligned} \tag{1}$$

donde $P_{j\leftarrow i}$ es la probabilidad de cambiar un ejemplo de la clase i a la clase j y K, el número total de clases. Este procedimiento tiende a balancear la distribución de las clases desequilibradas con el crecimiento del parámetro *fr*. La probabilidad de seleccionar un ejemplo de la clase mayoritaria es mayor que la de uno de la minoritaria.

Debido este hecho, la condición sobre el parámetro *fr* es menos restrictiva. Para conseguir que el ensemble converja, la mayoría de las instancias de cada clase deben permanecer intactas, manteniendo así la integridad de las clases. Generalmente, esto se satisface si $P_{j\leftarrow i} < P_{i\leftarrow i}$. Por tanto y de acuerdo con Eq. 1, la restricción sobre el parámetro *fr* se define como:

$$fr < (K-1)/K \tag{2}$$

La tendencia a equilibrar la distribución de las clases convierte a este ensemble en una propuesta inicial muy prometedora para lidiar con el problema de clasificación de conjuntos de datos no balanceados. Sin embargo, este método solo funciona correctamente con aquellos poco desequilibrados. En escenarios de alto desequilibrio, existe la posibilidad de inundar la clase minoritaria de ejemplos de la mayoritaria, perdiendo su integridad y comprometiendo la convergencia del ensemble. Además con el aumento del ratio de cambio para equilibrar las clases, se pueden perder la pocas instancias relevantes de la clase minoritaria. Por ello, es necesario un nuevo diseño que solucione dicho problema.

3. SwitchingNED: Intercambio de acuerdo a la Distancia al Enemigo más Cercano

En esta sección, se explica la propuesta SwitchingNED y cómo es capaz de tratar datos con alto desequilibrio de clases. Se detallan las diferentes partes

que conforma el algoritmo, desde las primeras decisiones hasta las últimas. Se justifican las decisiones tomadas, resolviendo, a su vez, los inconvenientes del Switching original hacia el problema. Así, se presenta la nueva manera de seleccionar las instancias basadas en la NED y la combinación con las técnicas de muestreo.

SwitchingNED mantienen intactas las instancias de la clase minoritaria, siendo solo las de la mayoritaria las que cambiarán. Este cambio permite retener toda la información relevante de la clase minoritaria. Dicha modificación es un paso importante para facilitar el proceso de balanceo para conjuntos de datos desequilibrados. Sin embargo, el problema de inundación con ruido sigue estando en aquellos altamente no balanceados, pudiendo perder importancia las instancias minoritarias reales.

Una manera de mitigar dicha perdida es asegurando que los ejemplos cambiados estén localizados lo más cerca posible de los ejemplos reales. Esto ayuda a los clasificadores a centrar su atención en los sub-espacios reales de la clase minoritaria, donde el número de instancias ha crecido.

Por ello, se ha diseñado una nueva manera de seleccionar las instancias a cambiar, siguiendo la idea anterior y desechando la manera aleatoria del Switching original. Las muestras de la clase mayoritaria son seleccionadas según su proximidad con las de la minoritaria. Concretamente, dicha proximidad se define como la Distancia al Enemigo más Cercano (NED), la distancia euclidiana a su ejemplo de la clase minoritaria más próximo. Tan lejos un ejemplo de la mayoritaria está de la clase minoritaria, mayor es la distancia a su enemigo más cercano, y consecuentemente, su selección es menos probable. Como se ha mencionado antes, dicho algoritmo se ha llamado SwitchingNED. Su implementación está representada en el Algoritmo 1.

SwitchingNED se aparta del concepto general del Switching original, al cambiar drásticamente la manera de seleccionar las instancias. La idea detrás del mecanismo original era introducir ruido controlado para generar diferentes perturbaciones de los datos de entrenamiento, mientras desaparece con el crecimiento del ensemble. Sin embargo, SwitchingNED usa las instancias intercambiadas para formar una mejor distribución de clases, manteniendo esa información en el comportamiento del ensemble. Por ello, una manera totalmente aleatoria de seleccionar las instancias seria muy perjudicial para dicho propósito.

El nuevo procedimiento de selección se ha implementado con una distribución de probabilidad calculada con la inversa de la NED de cada instancia mayoritaria. Dicha distribución es posteriormente utilizada para elegir los ejemplos mediante Selección por Ruleta en cada iteración, tal como se puede ver en la Línea 15. El Algoritmo 2 exhibe el proceso computacional de la distribución. Primero, se busca la instancia más cercana de la clase minoritaria para cada ejemplo de la mayoritaria. Su probabilidad es asignada como uno partido la distancia euclidiana a la instancia más cercana hallada. Finalmente, estas probabilidades se normalizan para que sumen uno. Este proceso es estático, calculado una vez al comienzo del algoritmo, como se muestra en la Línea 3.

Algoritmo 1 SwitchingNED con técnicas de muestreo.

function SWITCHING(D - datos, $nTrees$ - número de árboles, S - predicciones de D, fr - ratio de cambio, M - etiqueta mayoritaria, m - etiqueta minoritaria)
  inicializar: $S = \{\}$, $Arboles[1..nTrees]$, $\widehat{D}[1..nTrees]$, $Prob[1..size(D)]$
  $Prob = NEDprob(D, M, m)$
  if $fr < (P_M - P_m)/2$ **then**
    $newP_m = 0{,}5 - fr$
    $\#Cambiados = fr{*}N * P_m / newP_m$
    $\#Muestreados = N * P_M - (N * P_m + \#Cambiados)$
  else
    $\#Cambiados = fr{*}N$
    $\#Muestreados = 0$ ▷ Preprocesamiento no ejecutado
  end if
  for i in [1,$nTrees$] **do**
    $\widehat{D}[i] = D$
    for j in [1,$\#Cambiados$] **do** ▷ Switching Balancing stage
      $t = SeleccionRuleta(Prob)$
      $\widehat{D}[i][t]_{clase} = m$
    end for ▷ Preprocessing Balancing stage
    $\widehat{D}[i] = Muestreo(\widehat{D}[i], \#Muestreados, M)$
    $Arboles[i] = ConstruirArbol(\widehat{D}[i])$
  end for
  for d in D **do**
    $\widehat{S_{i,d}} = Predict(Arboles[i], d)$
  end for
  $S = PredecirMayorVoto(\widehat{S})$
  return S
end function

Algoritmo 2 NED probabilities function.

function NEDPROB(D - datos, M - etiqueta mayoritaria, m - etiqueta minoritaria)
  inicializar: $Prob[1..size(D)] = 0$, $Dist[1..size(D)]$
  for i in [1,$size(D)$] **do**
    if $D[i]_{clase} = M$ **then**
      for j in [1,$size(D)$] **do**
        if $D[j]_{clase} = m$ **then**
          if $distancia(D[i], D[j]) < Dist[i]$ **then**
            $Dist[i] = distancia(D[i], D[j])$
          end if
        end if
      end for
      $Prob[i] = 1/Dist[i]$
    end if
  end for
  $Normalizar(Prob[i])$
  return $Prob[i]$
end function

Como otros ensembles anteriormente, SwitchingNED se puede combinar con algoritmos estándares de preprocesamiento, para lidiar el problema de alto desequilibrio entre clases. Al balancear parcialmente con las técnicas de muestreo tradicional, se necesitan menos intercambios para equilibrar las clases. Este hecho reduce la posibilidad de inundación y facilitando la convergencia. Por otro lado, el uso de SwitchingNED como método de equilibrado permite reducir los posibles errores cometidos por los mecanismos de preprocesamiento, disminuyendo el número de instancias eliminadas de la clase mayoritaria (*under-sampling*) o las sintéticas de la minoritaria (*over-sampling*).

Esta combinación se consigue mediante infra-/sobre-muestrear hasta que SwitchingNED con un valor fijo del parámetro *fr* pueda igualar completamente la distribución de las clases. Es decir, si la distribución original es de 20 %-80 % y el valor de *fr* es 0.2, el algoritmo de muestreo cambiará las proporciones a 30 %-70 %, permitiendo a SwitchingNED completar la tarea cuando el 20 % de las instancias cambien a la clase minoritaria. SwitchingNED se puede combinar con cualquier técnica de muestreo, como Random Under-Sampling, Random Over-Sampling y SMOTE.

Tal como se ha diseñado, siempre se prioriza la selección de SwitchingNED a el método de muestreo. La distribución basada en NED se ejecuta independientemente al comienzo del algoritmo (Línea 3). Es decir, las probabilidades no se ven afectadas por la distribución posterior al muestreo. Además, las instancias son seleccionadas y cambiadas primero, como se expone en la Línea 14. Después, el muestreo se realiza sobre el resto de instancias (Línea 17). Si el valor de *fr* es suficiente para equilibrar las clases, el algoritmo de muestreo no se lanzará. Así, la ejecución de la técnica de muestreo está sujeta a la siguiente restricción:

$$fr < (P_M - P_m)/2 \tag{3}$$

donde P_M y P_m son la proporción de las clases mayoritaria y minoritaria, respectivamente. Véase en la Línea 4.

Debido a esto, *fr* se vuelve un parámetro bastante importante del algoritmo. Este establece, no solo la cantidad de instancias a cambiar, si no en qué grado el algoritmo de preprocesamiento toma parte en la fase de equilibrado (Líneas 4-11). Contra mayor es el valor de *fr*, menor es la relevancia del procedimiento de muestreo. *fr* debe de seleccionarse por el usuario, dependiendo de las necesidades del problema. Los experimentos previos al siguiente estudio experimental indican que, en media, lo mejores resultados de estos métodos son para *fr* igual a 0.1.

4. Marco experimental, Resultados y Análisis

En esta sección, se presenta el estudio experimental realizado. En 4.1, se comienza introduciendo el marco seguido, destacando los conjuntos de datos, medidas y parámetros usados. La sección 4.2 está dedicada a analizar el rendimiento de SwitchingNED con diferentes técnicas de muestreo. Por último, en 4.3, se comparará con el mejor ensemble hasta el momento, EUSBoost.

4.1. Marco experimental

El marco usado ha sido extraído completamente de [5], donde se realizó un completo estudio con los principales ensemble del estado del arte. Se ha usado exactamente los mismos 33 conjuntos de datos binarios altamente desequilibrados. Un conjunto se considera altamente desequilibrado cuando su Ratio de desequilibrio (IR), la cantidad de clase mayoritaria dividido de la minoritaria, es mayor que 9. La colección es muy diversa en términos de IR, desde casi sobrepasar 9 hasta exceder los cientos. Todos los conjuntos se han extraído de KEEL [1], siendo re-etiquetados de multi-clase estándar a conjuntos binarios no balanceados.

Se ha usado un esquema de 5 particiones de validación cruzada, con exactamente las mismas divisiones usadas en [5]. Cada ejecución se ha repetido 3 veces con diferentes semillas, por las propiedades aleatorias de los algoritmos. Como clasificador base de los ensembles se ha usado C4.5 [11], con el que se presentó Switching [3,9] y se ha usado en otros estudios similares [4,8], incluyendo [5]. Se ha construido 40 árboles en las propuestas SwitchingNED, por su similaridad en convergencia al Bagging [4]. Finalmente, como medida de evaluación, se ha usado el Área bajo la curva (AUC), ampliamente usada en dominios desequilibrados[4,8,10].

4.2. Evaluación de técnicas de muestreo en SwitchingNED

Se ha realizado este estudio para asegurar la mejor combinación con las técnicas de muestreo más populares (Random Under-sampling, Random Over-sampling and SMOTE): USwitchingNED, OSwitchingNED and SSwitchingNED. El tamaño del vecindario del SMOTE ha sido 5 y se ha usado la medida HVDM.

La Tabla 1 recoge los resultados AUC de las combinaciones exploradas, todos con un valor *fr* de 0.1. Se puede observar como USwitchingNED resulta ser bastante superior al resto, siendo en media y en la mayoría de conjuntos de datos, con la excepción de 12. El test estadístico de Wilcoxon también refleja ese dominio sobre el resto. En la Tabla 2, se observa como el rango obtenido por USwitchingNED es ampliamente mayor con resultados bastante bajos de los p-Valores.

4.3. USwitchingNED vs EUSBoost

Tras haber identificado la mejor propuesta, se ha analizado USwitchingNED con respecto al mejor ensemble de la literatura hasta el momento, EUSBoost [5], comprobando si la nueva propuesta es capaz de vencerlo. Se ha usado exactamente los parámetros recomendado y el marco teórico, donde se presentó [5].

La Tabla 3 recoge los resultados obtenidos en AUC de los 33 conjuntos. En esta, se aprecia una cierta superioridad del USwitchingNED sobre EUSBoost. De los 33, USwitchingNED resulta vencedor en 18, EUSBoost en 12 y en 3, ambos quedan empatados. En media, USwitchingNED sigue siendo el mejor método.

Cuadro 1: Resultados AUC para las propuestas diferentes de SwitchingNED.

	USwitchingNED	**SSwitchingNED**	**OSwitchingNED**
Glass04vs5	**0.9941**	**0.9941**	**0.9941**
Ecoli0346vs5	**0.9133**	0.9032	0.8939
Ecoli0347vs56	**0.9076**	0.9073	0.8694
Yeast05679vs4	0.8015	**0.8046**	0.7506
Ecoli067vs5	**0.9025**	0.8967	0.8758
Vowel0	0.9516	**0.9581**	0.9529
Glass016vs2	**0.7003**	0.6393	0.6138
Glass2	**0.7475**	0.7394	0.7109
Ecoli0147vs2356	0**.8872**	0.8716	0.8787
Led7Digit02456789vs1	0.8801	**0.9043**	0.8805
Ecoli01vs5	**0.9114**	0.8644	0.8598
Glass06vs5	**0.9950**	**0.9950**	**0.9950**
Glass0146vs2	0.7542	**0.7878**	0.7406
Ecoli0147vs56	**0.9096**	0.8889	0.8610
Cleveland0vs4	**0.8840**	0.7444	0.7975
Ecoli0146vs5	**0.9096**	0.8897	0.8519
Ecoli4	**0.9141**	0.8623	0.8357
Shuttle0vs4	**1.0000**	0.9997	0.9997
Yeast1vs7	**0.7782**	0.7592	0.6691
Glass4	**0.9001**	0.8958	0.8617
Page-blocks13vs4	0.9805	**0.9978**	**0.9978**
Abalone918	0.7189	**0.7294**	0.6368
Glass016vs5	**0.9714**	0.8943	0.8943
Shuttle2vs4	0.9918	**1.0000**	**1.0000**
Yeast1458vs7	0.5887	**0.5918**	0.5503
Glass5	0.9724	0.9451	**0.9878**
Yeast2vs8	0.7718	**0.7867**	0.7848
Yeast4	**0.8415**	0.7642	0.6887
Yeast1289vs7	**0.7020**	0.6479	0.6377
Yeast5	0.9569	**0.9740**	0.9439
Ecoli0137vs26	0.7935	**0.8354**	0.8348
Yeast6	**0.8808**	0.8144	0.8210
Abalone19	**0.6991**	0.5611	0.5339
Avg:	**0.8640**	0.8439	0.8244

Cuadro 2: Resumen del test Wilcoxon sobre las propuestas SwitchingNED.

Comparación	R^+	R^-	Hipótesis ($\alpha = 0{,}05$)	p-Valor
SSwitchingNEDvsOSwitchingNED	503.5	91.5	**Rejected**	0.00022
USwitchingNEDvsOSwitchingNED	527.0	68.0	**Rejected**	0.00003
USwitchingNEDvsSSwitchingNED	431.5	163.5	**Rejected**	0.02109

Sin embargo, las diferencias entre ambos no es suficiente para reflejar un rechazo de equivalencia del test Wilcoxon.

Cuadro 3: Resultados AUC para USwitchingNED y EUSBoost.

	EUSBoost	**USwitchingNED**
Glass04vs5	**0.9941**	**0.9941**
Ecoli0346vs5	0.8919	**0.9133**
Ecoli0347vs56	0.8842	**0.9076**
Yeast05679vs4	0.7869	**0.8015**
Ecoli067vs5	0.8767	**0.9025**
Vowel0	**0.9664**	0.9516
Glass016vs2	**0.7264**	0.7003
Glass2	0.7281	**0.7475**
Ecoli0147vs2356	0.8782	**0.8872**
Led7Digit02456789vs1	0.8755	**0.8801**
Ecoli01vs5	0.8848	**0.9114**
Glass06vs5	**0.9950**	**0.9950**
Glass0146vs2	**0.7768**	0.7542
Ecoli0147vs56	0.8941	**0.9096**
Cleveland0vs4	0.8164	**0.8840**
Ecoli0146vs5	0.8994	**0.9096**
Ecoli4	**0.9215**	0.9141
Shuttle0vs4	**1.0000**	**1.0000**
Yeast1vs7	0.7644	**0.7782**
Glass4	**0.9056**	0.9001
Page-blocks13vs4	**0.9869**	0.9805
Abalone918	0.7119	**0.7189**
Glass016vs5	**0.9886**	0.9714
Shuttle2vs4	**1.0000**	0.9918
Yeast1458vs7	**0.5948**	0.5887
Glass5	**0.9878**	0.9724
Yeast2vs8	0.7636	**0.7718**
Yeast4	0.8299	**0.8415**
Yeast1289vs7	**0.7184**	0.7020
Yeast5	0.9382	**0.9569**
Ecoli0137vs26	**0.8087**	0.7935
Yeast6	0.8601	**0.8808**
Abalone19	0.6730	**0.6991**
Avg:	0.8584	**0.8640**

En cambio, USwitchingNED tiene una clara ventaja en términos de complejidad computacional. La fase de entrenamiento del EUSBoost es mucho más costosa. Esto se debe a la ejecución del algoritmo genético EUS [6] en cada iteración del ensemble. EUS es muy costoso en tiempo, ya que cada evaluación requiere de un 1NN, repitiéndose hasta el número máximo dado (10000 por defecto). USwitchingNED carece de este sobre coste en cada iteración, requiriendo un 1NN al comienzo del algoritmo para el cálculo de la NED.

5. Conclusiones

En esta contribución, se ha propuesto un nuevo ensemble basado en Switching para clasificación altamente no balanceada. SwitchingNED es capaz de

aplicarse a dicho problema gracias al nuevo proceso de selección que incorpora basado en la Distancia al Enemigo más Cercano. Dicho procedimiento permite cambiar las instancias cercanas a las fronteras de la clase minoritaria, manteniendo así la integridad de la misma. Además, se ha diseñado una hibridación con las técnicas más populares de muestreo: Random Under/Over-sampling y SMOTE, resultando en USwitchingNED. Tras una comparativa con EUSBoost, se ha clarificado la bondad del método propuesto, pudiéndose considerar como uno de los mejores en este campo. En futuros estudios, se explorará cómo SwitchingNED afecta a los diferentes problemas que acompañan al desequilibrio de clases, tales como: la aparición de disjuntos, falta de densidad, solapamiento o ruido.

Referencias

1. Alcala-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17(2-3), 255–287 (2011)
2. Breiman, L.: Bagging predictors. Machine learning 24(2), 123–140 (1996)
3. Breiman, L.: Randomizing outputs to increase prediction accuracy. Machine Learning 40(3), 229–242 (2000)
4. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 42(4), 463–484 (2012)
5. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognition 46(12), 3460–3471 (2013)
6. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evolutionary computation 17(3), 275–306 (2009)
7. Lomax, S., Vadera, S.: A survey of cost-sensitive decision tree induction algorithms. ACM Computing Surveys 45(2), 16:1–16:35 (2013)
8. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences 250, 113 – 141 (2013)
9. Martínez-Muñoz, G., Suárez, A.: Switching class labels to generate classification ensembles. Pattern Recognition 38(10), 1483–1494 (2005)
10. Prati, R.C., Batista, G.E.A.P.A., Silva, D.F.: Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. Knowledge and Information Systems 45(1), 247–270 (2015)
11. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc. (1993)
12. Ramentol, E., Vluymans, S., Verbiest, N., Caballero, Y., Bello, R., Cornelis, C., Herrera, F.: Ifrowann: Imbalanced fuzzy-rough ordered weighted average nearest neighbor classification. IEEE Transactions on Fuzzy Systems 23(5), 1622–1637 (2015)