ISSN 1870-4069

Algoritmo de aprendizaje eficiente para tratar el problema del desbalance de múltiples clases

J. Monroy-de-Jesús, A. Guadalupe-Ramírez, J.C. Ambriz-Polo, E. López-González

Tecnológico de Estudios Superiores de Jocotitlán, Estado de México, México

gr.lizeth.a@gmail.com, juan_120990@hotmail.com, {juan.monroy, erika.lopez}@tesjo.edu.mx

Resumen. En este trabajo se presenta un enfoque de muestreo dinámico (I-SDSA) para tratar el problema de desequilibrio de múltiples clases. ISDSA es una modificación del algoritmo *Backpropagation*, que se enfoca en hacer un mejor uso de las muestras de entrenamiento para mejorar el rendimiento de clasificación del perceptrón multicapa (MLP). I-SDSA usa el error cuadrático medio y una función gaussiana para identificar las mejores muestras para entrenar la red neuronal. Los resultados que se muestran en este artículo destacan que I-SDSA aprovecha mejor el conjunto de datos de entrenamiento y mejora el rendimiento de clasificación de MLP. En otras palabras, I-SDSA es una técnica exitosa para lidiar con el problema del desequilibrio de múltiples clases. Además, los resultados presentados en este trabajo indican que el método propuesto es muy competitivo en términos de rendimiento de clasificación con respecto a los métodos clásicos de muestreo y otros enfoques dinámicos de muestreo, incluso en el tiempo de entrenamiento y el tamaño de la base de datos es mejor que en los métodos de muestreo.

Palabras clave: Backpropagation, MLP, múltiples clases, error cuadrático medio.

Efficient Learning Algorithm to Face the Multi-Class Imbalance Problem

Abstract. In this paper, a dynamic sampling approach (I-SDSA) is presented to deal with the problem of imbalance of multiple classes. I-SDSA is a modification of the Backpropagation algorithm, which focuses on making better use of training samples to improve multilayer perceptron (MLP) classification performance. I-SDSA uses the mean square error and a Gaussian function to identify the best samples to train the neural network. The results shown in this article highlight that I-SDSA takes better advantage of the training data set and improves the MLP classification performance. In other words, I-SDSA is a successful technique for dealing with the problem of imbalance of multiple classes. In addition, the results presented in this paper indicate that the proposed method is very competitive in terms of classification performance with respect to classical sampling methods

and other dynamic sampling approaches, even in the training time and the size of the base of data is better than in sampling methods.

Keywords: Backpropagation, MLP, multi-class, mean square error.

1. Introducción

El desbalance entre clases resulta ser un problema fundamental en las redes neuronales artificiales o ANN (Artificial Neural Network por sus siglas en inglés) ya que afecta el rendimiento del clasificador, no solo a su capacidad de generalización sino también en el costo computacional asociado a la fase de entrenamiento cuando éste es entrenado con métodos iterativos. El problema del desbalance de clases se presenta con mucha frecuencia en el mundo real, por ejemplo, en la detección de llamadas fraudulentas, imágenes de percepción remota, medicina, entre otras [1].

Se han propuesto diversos métodos para resolver el desbalance de clases, por ejemplo, el uso de técnicas de muestreo, los cuales duplican o eliminan patrones o muestras de entrenamiento hasta alcanzar un relativo equilibrio entre el número de muestras de las distintas clases (Over-sampling o Under-sampling) [2].

Uno de los métodos más comunes de sobre-muestreo (over-sampling), es la técnica de SMOTE (técnica de muestreo sintético) propuesta por Chawla et al. [3]. El funcionamiento de esta técnica consiste principalmente en generar nuevas muestras sintéticas interpoladas entre patrones de la clase minoritaria. Asimismo, el SMOTE ha servido de base para el desarrollo de otros métodos de sobre muestreo. Por ejemplo, *Bordeline-SMOTE, Adaptive Synthetic Sampling (ADASYN), SMOTE Editing Nearest Neighbor, Safe-LevelSMOTE, DBSMOTE, SMOTE + Tomeks Links*, entre otros.

Por otro lado, en el dominio de las técnicas de sub-muestreo (under-sampling), el método de sub-muestreo aleatorio ha sido reportado como uno de los más efectivos [4]. No obstante, se han generado numerosas aproximaciones a los métodos de sub-muestreo, las cuales se caracterizan por incluir un mecanismo heurístico en su funcionamiento. Básicamente este componente heurístico tiene como objetivo eliminar o cambiar las etiquetas de patrones ya sean ruido, atípicos o redundantes [5]. Por ejemplo, los métodos *Neighborhood Cleaning Rule, Onesided selection, Tomek links* y *Condensed Nearest Neighbor Rule*.

Actualmente, se está incrementado el interés por el estudio y desarrollo de métodos de muestreo dinámico, en los cuales la proporción y selección de muestras a duplicar o eliminar se realiza durante el entrenamiento del clasificador. Por ejemplo, en [6] se propone un nuevo algoritmo para determinar el nivel de equilibrio de clases, y además incluyen un mecanismo de selección de patrones de entrenamiento difíciles de aprender, con el propósito de mejorar la capacidad de generalización del MLP entrenado con el algoritmo Backpropagation.

Chawla et al. [7] propone un paradigma del tipo WRAPPER para determinar el nivel de sobre y sub-muestreo a aplicar en cada base de datos desbalanceada, que va ser usada para entrenar el clasificador. Wang and Jean [8] proponen el método SNOWBALL para entrenar redes del tipo MLP con datos desbalanceados, básicamente este método repite el entrenamiento de las muestras de las clases minoritarias hasta que el clasificador las identifica adecuadamente

En este sentido mientras que el problema de desbalance entre dos clases ha sido ampliamente estudiado, en dominios de múltiples clases, este inconveniente ha sido muy poco tratado.

2. Trabajos relacionados

En los últimos años el problema del desbalance de clases se ha abordado de muchas maneras y enfoques diferentes, sin embargo, los métodos más estudiados han sido los métodos de muestreo, por ejemplo, véanse las referencias [6, 9]. Estos métodos suelen ser eficaces y son independientes del clasificador.

Los métodos de muestreo pueden ser tan simples y claros como ROS o RUS [10], pero mientras que el primero replica muestras existentes en la clase minoritaria es más probable que ocurra sobre ajuste [11], y el segundo puede quitar tantas muestras permita la proporción de desbalance de clases, que, en algunos escenarios, sería inapropiado debido a la enorme pérdida de información en la base de datos.

Por lo tanto, se han desarrollado otros métodos de muestreo inteligentes que incluyen un mecanismo heurístico [12], como SMOTE, que crea muestras artificiales de la clase minoritaria mediante la interpolación de muestras existentes cerca de ellas [11] y de esta forma evitar la sobre especialización.

Otra técnica propuesta para superar las deficiencias de las técnicas de muestreo como ROS o SMOTE es Borderline-SMOTE [13], la cual selecciona muestras de la clase minoritaria que están en el límite, realizando sólo SMOTE en esas muestras. El muestreo sintético adaptativo (ADASYN) es una extensión de SMOTE, creando en la región límite más muestras entre las dos clases que en el interior de la clase minoritaria [14]. SMOTE Editing Nearest Neighbor (ENN) consiste en aplicar SMOTE y, a continuación, la regla ENN [15]. Safe-Level-SMOTE generan muestras de clase minoritaria sintéticas situadas más cerca del mayor nivel de seguridad, entonces todas las muestras sintéticas sólo se generan en regiones seguras [16].

SMOTE + Tomek Links (TL) [17] es la combinación de SMOTE y TL [15], Neighborhood Cleaning Rule usa la regla ENN, pero sólo elimina las muestras de la clase mayoritaria. Condensed Nearest Neighbor rule (CNN) [18] y One-sided selection eliminan las muestras redundantes, pero esta última usa TL. Show-Jane y Yue-Shi [19] presentan un nuevo método de sub-muestreo basado en métodos de agrupamiento para seleccionar los datos representativos como datos de entrenamiento para mejorar la precisión de clasificación para la clase minoritaria.

Otros enfoques de muestreo importantes se estudian en las referencias [9, 20, 21]. Se han propuesto métodos más sofisticados para tratar el problema del desbalance entre varias clases. Por ejemplo, el costo sensitivo (CS), que es uno de los temas más relevantes en la investigación en aprendizaje automático [22], es una buena solución para el problema de desbalance de clases [23].

El CS utiliza los costos asociados con la clasificación errónea de las muestras, emplea varias matrices de costos que definen los costos de clasificación errónea de cualquier muestra de datos [20]. Sin embargo, en estos métodos el costo de clasificación errónea debe ser conocido, pero en un problema de clasificación real, el costo de clasificación errónea es a menudo desconocido [24]. Zhi-Hua y Xu-Ying [22]

proporcionan un marco unificado para el uso de CS para abordar el desbalance de clases.

Los métodos ENSEMBLE es otro enfoque para tratar de resolver el problema de desbalance de clases. Esta técnica entrena múltiples componentes y luego combina sus predicciones [23, 25]. Sun et al. [24] emplean un conjunto de Máquinas de Vectores de Soporte, y el margen máximo se adopta para guiar el procedimiento de aprendizaje de conjuntos para la clasificación de imágenes de percepción remota. Galar et al. [26] presentan una revisión exhaustiva sobre los ensambles para el problema de desbalance de clases.

Recientemente, se han propuesto métodos de muestreo dinámico para resolver el problema de desbalance de múltiples clases. Estos establecen automáticamente la tasa de muestreo, por ejemplo, Fernández-Navarro et al. [27, 28] combinan métodos a nivel de datos con técnicas de entrenamiento dinámico. Utilizan algoritmos gen éticos para obtener la mejor relación de sobre-muestreo. Chawla et al. [7] proponen un paradigma Wrapper que descubre automáticamente la cantidad de sub-muestreo y tasa de sobre-muestreo para un conjunto de datos basado en optimización de las funciones de evaluación.

3. Análisis de muestras seguras, promedio y de frontera

En la literatura especializada sobre el problema de desbalance entre clases, se busca el interés de encontrar las mejores muestras para construir los clasificadores, eliminando aquellas muestras con alta probabilidad de ser ruido o muestras superpuestas [10, 29], es decir, aquellos cercanos a la decisión límite [13, 14] (este último ha sido menos explorado). Por lo tanto, en la literatura se pueden identificar básicamente tres categorías de muestras:

- Ruido y muestras raras o extrañas. Los primeros son casos con errores en sus etiquetas [30] o valores erróneos en sus rasgos que los describen, y los últimos son muestras minoritarias y raras situadas dentro de la clase mayoritaria [31].
- Las muestras fronterizas o superpuestas. Son aquellas localizadas donde se cruzan las regiones fronterizas de decisión [32].
- Las muestras seguras. Son aquellas con alta probabilidad de ser correctamente clasificadas y están rodeados de muestras de la misma clase [31].

Sin embargo, hay otras muestras que podrían ser de interés, las muestras situadas cerca de la decisión límite y lejos de las muestras seguras. Estas muestras se conocen como muestras" promedio".

En este trabajo en primera instancia se realizó la identificación de las muestras promedio, utilizando la salida de la red neuronal para analizar las muestras de entrenamiento, así como una función gaussiana γ, para identificar el tipo de muestra.

Esto se puede observar en las siguientes funciones:

$$\gamma(diff) = \exp(\frac{||diff - \mu||^2}{2\sigma^2}). \quad (1)$$

La variable $diff$ es la diferencia normalizada entre la salida real de la ANN para la muestra q:

$$diff = \frac{z_{min}^{q}}{\sqrt{(z_{min}^{q} - z_{maj}^{q})^2}} - \frac{z_{maj}^{q}}{\sqrt{(z_{min}^{q} - z_{maj}^{q})^2}}, \quad (2)$$

donde z_{min}^{q} y z_{maj}^{q} son las salidas reales de la ANN correspondientes a las clases minoritarias y mayoritarias (respectivamente) para una muestra q. La variable μ se calcula bajo la siguiente consideración: las salidas de la ANN se codifican usualmente en valores 0 y 1. Por ejemplo, para un problema de dos clases (clase A y clase B) las salidas ANN deseadas se codifican como (1; 0) y (0; 1)), respectivamente. Estos valores son las salidas objetivo de la ANN y los valores finales esperados son emitidos por la ANN después del entrenamiento. Por lo tanto, de acuerdo con este entendimiento, los valores esperados por μ son:

a) μ = 1.0 para muestras seguras, ya que se espera que la ANN se clasifique con alto nivel de precisión, las salidas de la ANN para todas las neuronas son valores cercanos a (0,1) o (1,0). Por lo tanto, si se aplica la Ecuación 2 el valor esperado (idealmente) es 1,0, por lo tanto, la función γ (Ecuación (1)) obtiene su valor máximo.
b) μ = 0.0 para muestras de frontera, ya que se espera que el clasificador no clasifique correctamente, es decir, las salidas esperadas para todas las neuronas son valores cercanos a (-0.5 o 0.5), por lo que en la ecuación (2) es aproximadamente 0.0, y la función γ (Ecuación 1) obtiene su valor máximo para estas muestras.
c) μ = 0.5 para muestras promedio, ya que se espera que la ANN se clasifique con menos exactitud, debido a que las muestras promedio se encuentran entre las muestras seguras (μ = 1.0) y frontera (μ = 0.0).

La función γ está propuesta para dar un cierto grado de prioridad a cada tipo de muestras. El objetivo es identificar cada tipo de muestra para ese valor μ. La ecuación. (2) da valores altos a las muestras cuando su *diff* (Ecuación (1)) es cercano a μ y valores bajos cuando el *diff* está lejos de μ.

Básicamente, el proceso para seleccionar las muestras es el siguiente: Antes de la formación de la ANN, el conjunto de datos de entrenamiento es equilibrado al 100% mediante una técnica eficaz de sobre-muestreo. Durante el entrenamiento, el método propuesto selecciona las muestras usando la ecuación (1) para actualizar los pesos de la red neuronal, elige desde el conjunto de datos de entrenamiento equilibrado sólo las mejores muestras para usar en el entrenamiento de la red neuronal.

3.1. Enfoque selectivo de muestreo dinámico (I-SDSA)

SDSA se basa en la idea de utilizar sólo las muestras más apropiadas durante la etapa de entrenamiento del MLP (muestras promedio), para mejorar el rendimiento del clasificador. I-SDSA funciona de la siguiente manera:

1. Antes del entrenamiento: Los datos de entrenamiento son balanceados al 100% mediante una técnica eficaz de sobre-muestreo.

2. Durante el entrenamiento: Del conjunto de datos de entrenamiento balanceados, I-SDSA elige las mejores muestras para ser usados en el entrenamiento del

MLP. Con el objetivo de identificar las mejores muestras, para esto utiliza la siguiente función:

$$\gamma(\Delta^q) = exp(-\frac{||\Delta^q - \mu||^2}{2\sigma^2}). \quad (3)$$

La variable Δq es la diferencia normalizada entre las salidas reales (Z), a y b de la red neuronal, para una muestra q:

$$\Delta^q = \frac{z_a^q}{\sqrt{(z_a^q - z_b^q)^2}} - \frac{z_b^q}{\sqrt{(z_a^q - z_b^q)^2}}, \quad (4)$$

$z\frac{a}{q}$ = max {zkq}; zbq = maxk=6 a {zkq} donde k = 1,2, 3..., K y K representan el número de clases en la base de datos. $z\frac{a}{q}$ y $z\,\frac{b}{q}$ son las dos salidas reales máximas de la red neuronal, correspondientes a una muestra q como se menciona en la ecuación (4).

Para seleccionar el valor de la variable μ, se aplica el siguiente proceso después de la iteración i: Obtener el nuevo MSE (MSE_i), si el $MSE_i < MSE_{(i-1)}$ se aplica la siguiente función, $\mu = \mu - \mu \cdot \epsilon$, donde $(0 < \epsilon < 1)$, y μ= 1 en la primera iteración (*i*=1); *i=1,2*, 3, I.

El enfoque selectivo de muestreo dinámico(I+SDSA) es detallado en el algoritmo 1.

Algoritmo 1 Enfoque selectivo de muestreo dinámico (I+SDSA) basado en el algoritmo estocástico Backpropagation.

Entrada: Datos de Entrenamiento **X**;

Salida: Pesos **W** y **U**;

INIT():

1: Leer archivo de configuración del MLP;
2: $i = 1, \mu = 1, \epsilon = 0.001$;
3: Generar pesos iniciales aleatorios entre −0.5 y 0.5;
LEARNING( ):
4: **while** (i < I) o (MSE_i > 0.0001) **do**
5: **for** q = 1 **to** Q **do**
6: $x^p \leftarrow$ elegir aleatoriamente una muestra **X;**
7: **FORWARD(**x^p**);**
8: $z_a^q = max_{(k=1,2,\ldots,K;)} \{z_k^q\}$;
9: $z_b^q = max_{(k=1,2,\ldots,K;\ k\neq a)} \{z_k^q\}$;
10: $\Delta^q = \frac{z_a^q}{\sqrt{(z_a^q - z_b^q)^2}} - \frac{z_b^q}{\sqrt{(z_a^q - z_b^q)^2}}$;
11: $\gamma(\Delta^p) = exp\ (-\ ||\Delta^q - \mu||^2/2\sigma^2)$;
12: **if** (**Random( )** <= $\gamma(\Delta^q)$) **then**
13: **UPDATE(**x^p**);**
14: **end if**

```
15:  end for
16: if (MSE_i < MSE_(i-1) and i > 1) then
17:     μ = μ − μ · ϵ;
18:  end if
19: i + +;
20: end while
```

Para la etapa experimental se usaron quince conjuntos de datos, que fueron obtenidas de cinco bases de datos de percepción remota procedentes del mundo real, (Cayo, Feltwell, Satimage, Segment y 92AV3C).

Los conjuntos de datos originales fueron alterados uniendo y/o reduciendo al azar el tamaño de algunas clases con el fin de obtener conjuntos de datos de múltiples clases desbalanceadas con varias clases de distribución.

El conjunto de datos 92AV3C utilizado en este trabajo es una versión reducida del conjunto de datos original con seis clases (2, 3, 4, 6, 7 y 8) y treinta y ocho atributos. La Tabla 1 muestra las principales características de este proceso.

Se aplicó el método de validación cruzada de diez veces a todos los conjuntos de datos empleados en el proceso experimental.

Para entrenar al MLP se utilizó el Backpropagation y cada proceso de entrenamiento se realizó diez veces, en otras palabras, los pesos fueron iniciados aleatoriamente diez veces.

Se eligió el algoritmo estocástico Backpropagation, ya que suele ser mucho más rápido y a menudo resulta en mejores soluciones que el Backpropagation por lotes, y se puede utilizar para el seguimiento de los cambios [33], que permite directamente aplicar el mecanismo de selección del método propuesto (ecuación (3)).

La razón de aprendizaje (η) se fijó en 0.1, y se estableció el criterio de detención a 500 iteraciones o si el valor MSE es inferior a 0.001. Se utilizó una sola capa oculta y para el número de neuronas en la capa oculta para cada conjunto de datos se estableció mediante prueba y error. El número de neuronas se fijaron en: 7 para MCAA, MCAB y MCAC; a 6 para MFEA, MFEB y MFEC; a 12 para MSAA, MSAB y MSAC; a 10 para MSEA, MSEB, MSEC, M92A, M92B y M92C. La variable *k* en SMOTE se estableció en cinco como en [11] ya que fue un trabajo que se tomó como referencia de comparación.

Para evaluar y comparar el rendimiento clasificador del método propuesto y los otros enfoques, se utilizó una versión para problemas de múltiples clases (ver ecuación (5)) del área bajo la curva, característica del receptor (MAUC) [34], es un método para la validación de clasificadores en escenario de desbalance de múltiples clases. El MAUC es define como:

$$MAUC = \frac{2}{\|J\|(\|J\|-1)} \sum_{j_i, j_k \epsilon J} AUC_R(j_i, j_k) \quad (5)$$

donde $AUC_R(j_i \; y \; j_k)$ es el AUC para cada par de clases $j_i \; y \; j_k$.

Tabla 1. Resumen de las principales características del conjunto de datos.

Base de Datos	#Ejemplos.	#Atributos.	# Ejemplos por clases						
			1	**2**	**3**	**4**	**5**	**6**	**7**
MCAA	6019	4	2941	293	2283	369	133	–	–
MCAB			3310	293	2283	133	–	–	–
MCAC			3074	293	2283	369	–	–	–
MFEA	8536	15	3531	2441	91	2295	178	–	–
MFEB			5972	178	91	2295	–	–	–
MFEC			5826	2441	91	178	–	–	–
MSAA	4697	36	1508	1533	104	1358	93	101	–
MSAB			3041	101	104	1358	93	–	–
MSAC			2866	1533	104	101	93	–	–
MSEA	1470	19	330	50	330	330	50	50	330
MSEB			660	50	330	330	50	50	–
MSEC			660	50	330	330	50	50	–
M92A	5063	38	190	117	1434	2468	747	106	–
M92B			190	117	1434	3215	106	–	–
M92C			190	117	3902	106	747	–	–

Por otro lado, con el fin de fortalecer los resultados del análisis, se aplicaron las pruebas estadísticas no paramétricas de Friedman e Iman - Davenport, para saber si existe diferencia estadística significativa en los resultados.

Finalmente, cuando existe alguna diferencia significativa entre los métodos individuales utilizados, se aplicó las pruebas post hoc de Holm [35] y Shaffer [36] con el fin de encontrar el par de métodos particulares que producen diferencias estadísticas significativas. En las referencias. [37,38] se presenta un estudio exhaustivo de estos métodos estadísticos no paramétricos. Se emplearon las pruebas de Friedman, Iman - Davenport, Holm y Shaffer con $\alpha = 0.05$ para el nivel de confianza, esto con ayuda del software KEEL [39].

4. Resultados

La experimentación se basó principalmente en la comparación con diferentes métodos. En primer lugar, I-SDSA se compara con métodos paralelos (SDSA y DyS). En segundo lugar, con tres enfoques de muestreo convencionales (SMOTE, ROS y RUS), que han demostrado su capacidad para tratar el problema de desbalance de clases

BD	I-SDSA-S	SDSA-R	SMOTE	SDSA-S	ROS	I-SDSA-R	DYS	DOS	RUS	STANDARD
MCAA	0.897	0.896	**0.900**	0.897	0.896	0.894	0.869	0.837	0.860	0.689
MCAB	0.923	0.924	0.926	**0.930**	0.925	0.923	0.908	0.910	0.891	0.687
MCAC	0.904	0.902	0.905	0.903	**0.907**	0.903	0.854	0.890	0.870	0.742
MFEA	**0.935**	0.927	0.929	0.927	0.928	0.930	0.927	0.917	0.902	0.780
MFEB	0.945	0.935	0.941	**0.949**	0.937	0.940	0.930	0.915	0.919	0.729
MFEC	**0.932**	0.928	0.929	**0.932**	0.912	0.913	0.922	0.928	0.910	0.750
MSAA	0.832	0.834	0.835	0.830	0.827	0.821	0.824	0.831	0.815	0.734
MSAB	0.840	0.824	0.841	**0.852**	0.813	0.816	0.823	0.829	0.826	0.686
MSAC	**0.864**	0.838	**0.864**	0.853	0.830	0.841	0.823	0.848	0.798	0.710
MSEA	0.945	**0.958**	0.943	0.946	0.951	0.945	0.949	0.945	0.923	0.922
MSEB	0.949	**0.951**	0.943	0.942	0.945	0.943	0.950	0.944	0.915	0.916
MSEC	0.941	0.943	0.935	0.930	0.938	0.941	0.937	0.935	0.912	0.910
M92A	0.842	0.866	0.837	0.814	0.859	0.857	0.861	0.796	0.773	0.734
M92B	0.849	0.860	0.846	0.845	0.868	0.863	0.857	0.789	0.803	0.715
M92C	0.889	0.911	0.885	0.898	0.906	0.906	0.924	0.887	0.887	0.819
MAUC	**0.899**	**0.900**	**0.897**	**0.897**	**0.896**	**0.896**	**0.891**	**0.880**	**0.867**	**0.768**
Ranking	**4.300**	**4.567**	**4.833**	**5.233**	**5.300**	**6.200**	**6.800**	**7.900**	**10.167**	**12.367**

Fig. 1. Rendimiento de la clasificación del Backpropagation usando MAUC. Los números en negrita representan los mejores valores.

[9, 11, 21], prácticamente son métodos que balancean las muestras antes de la clasificación.

La comparación se desarrolló a partir de tres enfoques: 1) rendimiento de clasificación, 2) muestras utilizadas en el entrenamiento, y 3) tiempo de entrenamiento. De la misma forma, se incluyeron pruebas estadísticas no paramétricas para informar cuando existe diferencia estadística significativa en los resultados.

La Figura 1 muestra los promedios MAUC y rangos Friedman, obtenidos en la etapa de clasificación de los métodos estudiados. Friedman clasifica el método establecido en el rango 1 al mejor algoritmo, 2 en el segundo mejor, 3 en el tercer mejor, sucesivamente para todos los casos; si existen semejanzas, se calcula el rango promedio [37, 38].

Por otra parte, en la Figura 1 se observa que todos los métodos estudiados mejoran el rendimiento de la clasificación (considerando como referencia los resultados estándar del Backpropagation (STANDARD) y la familia SDSA (ISDSA-R, I-SDSAS, SDSA-R, SDSA-S), los métodos de sobre-muestreo (ROS y SMOTE) producen prácticamente los mismos resultados. Por lo tanto, de acuerdo con los rangos de Friedman, I-SDSA-S presenta una tendencia a obtener mejores resultados que los otros métodos, pero SDSA-R en términos de promedio de MAUC es mejor que I-SDSA-S.

Sin embargo, SDSA-S y SDSA-R necesitan un modelo independiente de validación para funcionar correctamente, prueba diferentes valores de μ para obtener el mejor. Para la experimentación se utilizaron los siguientes valores de μ para SDSA-S y SDSA-R: 0.125, 0.25, 0.375, 0.5, 0.625, 0.75 y 0.875.

Para emplear una validación independiente, se involucraron dos problemas: 1) ¿Necesita costo computacional adicional, y 2) Cuantos valores μ diferentes necesitamos probar? En I-SDSA-S se obtiene automáticamente el valor apropiado de μ para cada conjunto de datos durante el entrenamiento de MLP. Para ello se utilizó el MSE del Backpropagation.

De la misma forma, podemos ver en la Figura 1 que I-SDSA presenta un comportamiento opuesto a SDSA, por ejemplo, I-SDSA-S es mejor que I-SDSA-R y SDSA-S es peor que SDSA-R. La explicación de esto es precisamente que SDSAR y

SDSA-S aplican para cada conjunto de datos el mismo valor en (μ = 0. 125), mientras que I-SDSA utiliza diferentes valores para cada conjunto de datos. En el mismo sentido en la Figura 1, se observa que, SMOTE es mejor que ROS. Por otro lado, SDSA es consistente con los resultados de la Referencia [40], es decir, para utilizar ROS como método de sobre-muestreo en SDSA es mejor que aplicar SMOTE.

DyS y DOS no mejoran los resultados de la familia SDSA, mucho menos ROS y SMOTE, sin embargo, sus resultados no son malos en sí mismos. ROS y SMOTE son efectivos para tratar el problema de desbalance de clases, sin embargo, SMOTE y ROS necesitan más muestras (Fig. 1) y tiempo (Fig. 2) en la etapa de entrenamiento que los otros métodos estudiados. La principal ventaja de estos métodos de sobre-muestreo es que son los más simples. La tendencia de RUS es obtener mejores resultados que el STANDARD, sin embargo, su desempeño es peor que los otros métodos. El bajo rendimiento de clasificación de RUS podría explicarse por el número de muestras borradas en el conjunto de datos de formación para este método, debido a la perdida de información pertinente (Fig.1).

Por otro lado, la Fig. 1 muestra que la familia DyS y SDSA hacen un mejor uso de las muestras de entrenamiento, es decir, no requieren todas las muestras de entrenamiento. La familia SDSA utiliza menos muestras que STANDARD, pero la familia SDSA gasto aproximadamente 50% más de tiempo en la etapa de entrenamiento que la STANDARD (ver Fig. 2). Esto se debe a que la familia SDSA no elimina ninguna muestra durante el entrenamiento. Por el contrario, DyS utiliza considerablemente menos muestras (Fig. 1) y paso mucho menos tiempo en la etapa de entrenamiento (al rededor del 50% de las muestras de entrenamiento y el tiempo con respecto al STANDARD, ver Figura. 2).

Sin embargo, de acuerdo con la Tabla 2, DyS no muestra una tendencia a superar el rendimiento de clasificación de los métodos de la familia ROS, SMOTE y SDSA. Por su parte, el DOS emplea menos muestras y gasto menos tiempo que ROS y SMOTE, pero necesita más muestras y tiempo que DyS y SDSA

(Fig. 1 y 2). Con respecto al rendimiento de clasificación, DOS es solo mejor que RUS y STANDARD. Su principal inconveniente es que utiliza el número de iteraciones como parámetro para actualizar la ratio de desbalance, por lo que el rendimiento del clasificador se vincula a este número. En [41] los autores utilizan 5000 ´épocas en el entrenamiento, en la experimentación del desarrollo del trabajo se aplican solo 500, por esta razón DOS usa más muestras en la etapa de entrenamiento que en la obra original.

Con el fin de reforzar el análisis de resultados se aplicó un análisis estadístico no paramétrico y considerando el rendimiento de reducción, se distribuye de acuerdo con la chi-cuadrado con 13 grados de libertad, la estadística de Friedman se establece en 100.638, y valor de *p* calculado por la prueba de Friedman es de 7.432E-11.

Considerando el rendimiento de reducción distribuido según la distribución *F* con 13 y 143 grados de libertad, la estadística de Iman y Davenport nos proporciona un valor de 19.996 y el valor *p* calculado por la prueba de Iman-Davenport es de -2.220E-16. Por lo tanto, la hipótesis nula es rechazada, es decir, las pruebas de Friedman e Iman-Davenport indican que existen diferencias significativas en los resultados.

Por otra parte, se utilizaron los procedimientos estadísticos de Holm y Shaffer para realizar el análisis estadístico no paramétrico post-hoc. Este análisis se realiza para averiguar qué algoritmos son diferentes entre todas las comparaciones $C \times C$ de los clasificadores *C*.

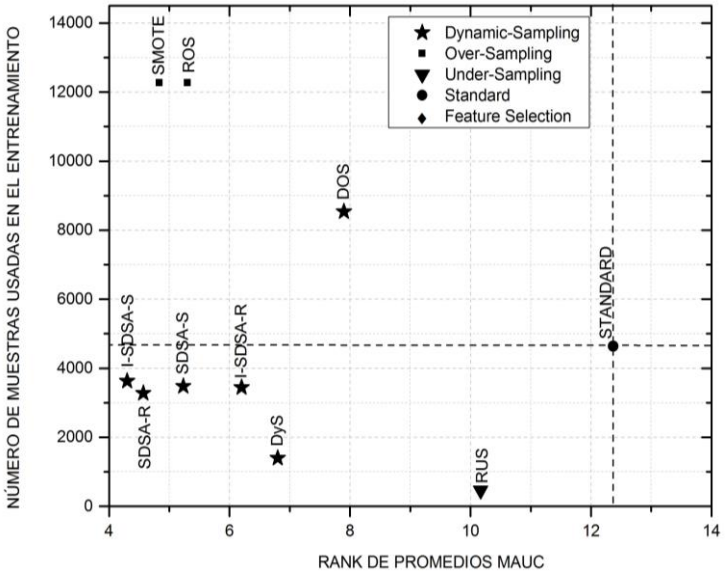

Fig. 1. Comparación de rangos medios MAUC versus muestras utilizadas en el entrenamiento. El tamaño se considera con referencias al tamaño de la base de datos original.

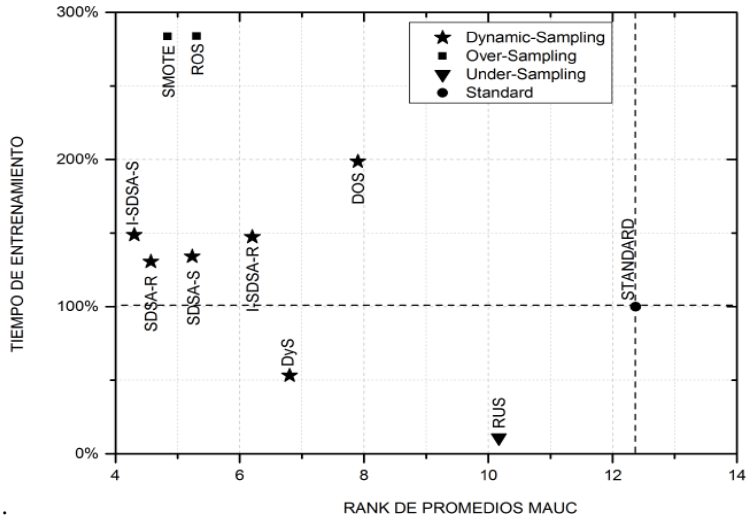

Fig. 2. Comparación de rangos medios de MAUC versus tiempo de entrenamiento. El tiempo (100%) corresponde al tiempo usado en la etapa de entrenamiento del Backpropagation estándar con el conjunto de datos original.

La Tabla 3. Presenta los valores de p no ajustados y los valores de p ajustados (α) para los procedimientos estadísticos de Holm (p-Holm) y Shaffer (p-Shaffer) considerando un α= 0.05.

Tabla 3. Valores de *P* ajustados y no ajustados para comparaciones de CXC, sobre 15 bases de datos, teniendo en cuenta un valor de efectividad de α= 0.05.

	STANDART	SMOTE	SDSA-S	SDSA-R	RUS	ROS	I-SDSA-S	I-SDSA-R	DyS	DOS
STANDARD		**0.000002**	**0.000009**	**0.000006**	0.11842	**0.000018**	**0.000001**	**0.000116**	0.00139	0.01373
		0.000602	**0.000685**	**0.000667**	0.00122	**0.000704**	**0.000581**	**0.000735**	0.00079	0.00091
		0.000641	**0.000746**	**0.000746**	0.00122	**0.000746**	**0.000641**	**0.000746**	0.00079	0.00091
SMOTE	0.73268		0.78841	0.00128	0.62559	0.94165	0.35387	0.11278	0.02046	
	0.00278		0.00333	0.00078	0.00208	0.01250	0.00152	0.00119	0.00094	
	0.00278		0.00333	0.00078	0.00208	0.01250	0.00152	0.00119	0.00094	
SDSA-S	0.94165	0.00399	0.88362	0.67832	0.55819	0.21340	0.04813			
	0.01667	0.00085	0.00625	0.00227	0.00192	0.00139	0.00104			
	0.16667	0.00085	0.00625	0.00227	0.00192	0.00139	0.00104			
SDSA-R	0.00316	0.82620	0.73268	0.51007	0.18768	0.04042				
	0.00082	0.00357	0.00263	0.00179	0.00132	0.00100				
	0.00082	0.00357	0.00263	0.00179	0.00132	0.00100				
RUS	0.00629	0.00099	0.02183	0.10213	0.36668					
	0.00088	0.00077	0.00096	0.00116	0.00156					
	0.00088	0.00077	0.00096	0.00116	0.00156					
ROS	0.57470	0.66055	0.27225	0.67278						
	0.00200	0.00217	0.00143	0.00109						
	0.00200	0.00217	0.00143	0.00109						
I-SDSA-S	0.31717	0.09711	0.01681							
	0.00147	0.00114	0.00093							
	0.00147	0.00114	0.00093							
I-SDSA-R	0.51007	0.16433								
	0.00172	0.00128								
	0.00172	0.00128								
DyS	0.46421									
	0.00167									
	0.00167									
DOS										

Las filas y columnas representan los métodos estudiados. Los valores *p* no ajustados y los métodos de *p*-Holm y *p*-Shaffer. Para cada método se muestran tres valores de *p*, el primero es el valor *p* no ajustado, el segundo es de *p*-Holm, y el ultimo es el valor de *p*-Shaffer.

El procedimiento de Holm rechaza aquellas hipótesis que tienen un valor *p* no ajustado ≤ *p*-Holm, y el procedimiento de Shaffer rechaza aquellas hipótesis que tienen un valor *p* no ajustado ≤ p-Shaffer. La hipótesis nula rechazada se escribe en negrita. En las Tablas 2 y 3 se observa que los métodos de ROS, SMOTE, DyS y SDSA clasifican mejor (con estadística significativa) que el ESTANDAR. ´ Con estadística significativa en sus resultados I-SDSA-S, SDSA-R y SMOTE, mejoran el rendimiento de clasificación que de RUS y ROS.

DOS y DyS presentan un rendimiento inferior al de ROS, SMOTE y SDSA, pero de acuerdo con la prueba post-hoc de Holm-Shaffer, sus resultados de clasificación no son estadísticamente diferentes.

ROS, SMOTE y SDSA son los mejores métodos estudiados en este trabajo, no obstante, sus resultados no muestran diferencia con significancia estadística entre ellos.

5. Conclusiones y trabajos futuros

Los resultados mostrados demuestran que I-SDSA es muy competitivo en el desempeño de la clasificación con respecto a los métodos de sobre-muestreo y sub-muestreo (ROS, SMOTE y RUS), y con enfoques similares como los métodos de muestreo dinámico (DyS) o SDSA.

I-SDSA es mejor en términos de muestras de entrenamiento, tiempo de entrenamiento y desempeño de clasificación.

DyS y RUS necesitan menos muestras, incluso menos tiempo que I-SDSA, pero la tendencia es que el desempeño de clasificación del método propuesto debe ser mejor. I-SDSA usa menos muestras que el Backpropagation estándar, pero requiere un 50% más de tiempo de entrenamiento, en este sentido I-SDSA es un enfoque exitoso para abordar el problema de desbalance de varias clases, porque hace un mejor uso de las muestras de entrenamiento que permite mejorar el desempeño de la clasificación. En conclusión, el algoritmo presentado en este trabajo (I-SDSA) es una buena estrategia para tratar el desbalance entre varias clases, ya que hace un mejor uso de las muestras de entrenamiento que permitiendo mejorar el desempeño no de la clasificación.

Como trabajos futuros se pretende usar aprendizaje profundo (Deep Lear Ning) para trabajar con bases de datos de gran tamaño (bit data) y mejorar los tiempos de entrenamiento.

Referencias

1. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. In: Emerging Artificial Intelligence Applications in Computer Engineering, pp. 3–24 (2007)
2. Batista, G., Prati, R., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. (SIGKDD) Explor. Newsl., 6, pp. 20–29 (2004)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W. P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res., 16, pp. 321–357 (2002)
4. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis, 6, pp. 429–449 (2002)
5. Batista, G., Silva, D., Prati, R: An experimental design to evaluate class imbalance treatment methods. In: International Conference on Machine Learning and Applications (ICMLA), 2, pp. 95–101 (2012)
6. Lin, M., Tang, K., Yao, X.: Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Trans. Neural Netw. Learning Syst., 24(4), pp. 647–660 (2013)
7. Chawla, N., Cieslak, D., Hall, L., Ajay, J.: Automatically countering imbalance and its empirical relationship to cost. Data Min. Knowl. Discov., 17, pp. 225– 252 (2008)
8. Wang, J., Jean, J.S.N.: Resolving multifont character confusion with neural networks. Pattern Recognition, 26(1), pp. 175–187 (1993)
9. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling techniques. IEEE Transactions on Knowledge and Data Engineering, pp. 1041–4347 (2015)

10. Batista, G., Prati, R., Monard, M.: Balancing strategies and class overlapping. IDA, pp. 24–35 (2005)
11. Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, vol. 16, pp. 321–357 (2002)
12. Prati, R.C., Batista, G., Monard, M.C.: Data mining with imbalanced class distributions: concepts and methods. In: Proceedings of the 4th Indian International Conference on Artificial Intelligence, (IICAI), pp. 359–376 (2009)
13. Han, H., Wang, W., Mao, B.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. (ICIC), pp. 878–887 (2005)
14. He, H., Bai, Y., Garcia, E., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. (IJCNN), pp. 1322–1328 (2008)
15. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. (SIGKDD), Explor. Newsl., 6, pp. 20–29 (2004)
16. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD09), 5476 of Lecture Notes on Computer Science, pp. 475–482, Springer-Verlag (2009)
17. Tomek, I.: Two Modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, 7(2), pp. 679–772 (1976)
18. Hart, P.: The condensed nearest neighbour rule. IEEE Transactions on Information Theory, 14(5), pp. 515–516 (1968)
19. Show-Jane, L., Yue-Shi, Y.: Cluster-based under-sampling approaches for imbalanced data distributions. Expert Systems with Applications, 36, pp. 5718–5727 (2009)
20. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), pp. 1263–1284 (2009)
21. 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. Inf. Sci., 250, pp. 113–141 (2013)
22. Zhou, Z.H., Liu, X.Y.: On multi-class cost-sensitive learning. Computational Intelligence, 26(3), pp. 232–257 (2010)
23. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 18, pp. 63–77 (2006)
24. Sun, T. Jiao, L., Feng, J., Liu, F., Zhang, X.: Imbalanced hyperspectral image classification based on maximum margin. IEEE Geosci. Remote Sensing Lett., 12(3), pp. 522–526 (2015)
25. Mirza, B., Lin, Z., Liu, N.: Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. Neurocomputing, 149, pp. 316–329 (2015)
26. Galar, M., Fernández, A., Tartas, E.B., Sola, H.B., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybridbased approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 42(4), pp. 463–484 (2012)
27. Fernández-Navarro, F., Hervás-Martínez, C., Gutiérrez, P.A.: A dynamic oversampling procedure based on sensitivity for multi-class problems. Pattern Recogn., 44, pp. 1821–1833 (2011)
28. Fernández-Navarro, F., Hervás-Martínez, C., García-Alonso, C., Torres-Jiménez, M.: Determination of relative agrarian technical efficiency by a dynamic over-sampling procedure guided by minimum sensitivity. Expert Syst. Appl., 38(10), pp. 12483–12490 (2011)
29. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine, (AIME), pp. 63–66, Springer Verlag (2001)

30. 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, pp. 113–141 (2013)
31. Stefanowski, J.: Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In: Emerging Paradigms in Machine Learning, S. Ramanna, L. C. Jain, and R. J. Howlett (eds.), 13 of Smart Innovation, Systems and Technologies, pp. 277–306, Springer Berlin Heidelberg (2013)
32. Alejo, R., Valdovinos, R., García, V., Pacheco-Sanchez, J.H.: A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. Pattern Recognition Letters, 34(4), pp. 380–388 (2013)
33. Lecun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient BackProp. In: Neural Networks—Tricks of the Trade, G. Orr and K. Müller (eds.), 1524 of Lecture Notes in Computer Science, pp. 5–50, Springer Verlag (1998)
34. Fawcett, T.: An introduction to roc analysis. Pattern Recogn. Lett., 27, pp. 861–874 (2006)
35. Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6(2), pp. 65–70 (1979)
36. Shaffer, J.: Modified sequentially rejective multiple test procedures. Journal of the American Statistical Association, 81(375), pp. 826–831 (1986)
37. García, S., Herrera, F.: An Extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all Pairwise Comparisons. Journal of Machine Learning Research, 9, pp. 2677–2694 (2008)
38. Luengo, J., García, S., Herrera, F.: A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests. Expert Systems with Applications, 36(4), pp. 7798–7808 (2009)
39. Alcalá-Fernández, J., Fernandez, 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), pp. 255–287 (2011)
40. Alejo, R., Monroy-de Jesús, J., Pacheco-Sánchez, J., López-González, H., Antonio-Velázquez, J. A.: A selective dynamic sampling back-propagation approach for handling the two-class imbalance problem. Applied Sciences, 6(7), pp. 200 (2016)
41. Alejo, R., García, V., Pacheco-Sánchez, J. H.: An efficient over-sampling approach based on mean square error back-propagation for dealing with the multiclass imbalance problem. Neural Processing Letters, pp. 1–16 (2015)