

Tratamiento del desbalance en problemas con múltiples clases con ECOC

Rosa María Valdovinos Rosas¹, Rosalinda Abad Sánchez, Roberto Alejo Eleuterio²,
Edgar Herrera Arteaga^{1,3} y Adrián Trueba Espinosa⁴

¹ Universidad Autónoma del Estado de México, Facultad de Ingeniería,
Ciudad Universitaria, Toluca, México

² Tecnológico de Estudios Superiores de Jocotitlán,
México

³ Instituto Nacional de Investigación Nuclear ININ, La Marquesa, Ocoyoacac,
México

⁴ Centro Universitario UAEM Texoco,
México

{li_rmvr, ralejoll}@hotmail.com, edgar.herrera@inin.gob.mx

Resumen. El problema del desbalance de clases puede producir un deterioro importante en la efectividad del clasificador, en particular con los patrones de las clases menos representadas. El desbalance en el conjunto de entrenamiento (CE) significa que una clase es representada por una gran cantidad de patrones mientras que otra es representada por muy pocos. Los estudios existentes se encuentran orientados principalmente a tratar problemas de dos clases, no obstante, un importante número de problemas reales se encuentran representados por múltiples clases, donde resulta más difícil su discriminación para el clasificador. El éxito de la Mezcla de Expertos (ME) se basa en el criterio de “divide y vencerás”. En su funcionamiento general, el problema es dividido en fragmentos más pequeños que serán estudiados por separado. De este modo, el modelo general es poco influenciado por las dificultades individuales de sus componentes. La idea principal del estudio aquí mostrado, es construir una Mezcla de expertos cuyos miembros serán entrenados en una parte del problema general y de este modo, mejorar el rendimiento del clasificador en el contexto de múltiples clases. Para este fin, se hace uso de los métodos conocidos como *Error-correcting output codes* (ECOC), que permiten realizar una codificación en parejas de clases el problema de estudio. Resultados experimentales sobre conjuntos de datos reales, muestran la viabilidad de la estrategia aquí propuesta.

Palabras clave. Desbalance de clases, mezcla de expertos, fusión, múltiples clases, error correcting output codes (ECOC).

Handling the Multi-Class Imbalance Problem using ECOC

Abstract. Imbalanced training sample means that one class is represented by a large number of examples while the other is represented by only a few. This problem may produce an important deterioration of the classifier performance, in particular with patterns belonging to the less represented classes. The majority of the studies in this area are oriented, mainly, to resolve problems with two classes. However, many real problems are represented by multiple classes, where it is more difficult to discriminate between them. The success of the Mixture of Experts (ME) strategy is based on the criterion of “divide and win”. The general process divides the global problem into smaller fragments which will be studied separately. In this way, the general model has few influences of the individual difficulties (of their members). In this paper we propose a strategy for handling the class imbalance problem for data sets with multiple classes. For that, we integrate a mixture of experts whose members will be trained as a part of the general problem and, in this way, will improve the behavior of the whole system. For dividing the problem we employ the called *Error-correcting output codes* (ECOC) methods, when the classes are codified in pairs, which are considered for training the mixture of experts. Experiments with real datasets demonstrate the viability of the proposed strategy.

Keywords. Class imbalance, fusion, mixture of experts, error correcting output codes (ECOC).

1 Introducción

Una Mezcla de expertos (ME) es un método en el cual múltiples clasificadores son entrenados individualmente y combinan sus decisiones individuales para obtener la decisión general del sistema [7]. La idea fundamental [6] considera la utilización de un grupo de clasificadores $D = \{D_1, \dots, D_n\}$, donde cada clasificador tendrá como entrada un vector de atributos $\in \mathbb{R}^n$, al cual se asigna una etiqueta de clase. En la fase final, las decisiones individuales de la ME son combinadas mediante algún esquema de toma de decisiones para determinar la clase definitiva que se asigna al patrón y.

Algunos aspectos que estos sistemas pretenden optimizar y que están presentes al utilizar un único clasificador son [11]: la decisión combinada toma ventaja sobre las decisiones individuales de cada clasificador, los errores correlacionados de los componentes individuales pueden ser eliminados cuando se considera el total de las decisiones, los patrones de entrenamiento y que pueden no proporcionar información suficiente para seleccionar el mejor clasificador, el algoritmo de aprendizaje puede no ser adaptado para resolver el problema y, finalmente, el espacio individual de búsqueda puede no contener la función objetivo.

De forma general, la ME se construye en dos etapas: entrenamiento de los clasificadores individuales y combinación de las decisiones individuales. Al respecto, en la construcción de los clasificadores individuales, es importante considerar el grado de diversidad, ya sea en la naturaleza de la regla de decisión o la información proporcionada a cada uno y, finalmente, en la toma de decisiones.

En lo concerniente a la construcción de los miembros individuales de la ME, la literatura distingue métodos que generan varias submuestras mediante la selección de patrones, atributos o clases [7]. Por otro lado, para obtener la decisión final de la ME, en la literatura se proponen dos estrategias para realizar la combinación de las decisiones individuales de los clasificadores: la fusión y la selección. En la primera se asume que todos los clasificadores son competitivos y complementarios (igualmente

expertos), en tanto que la selección se elige sólo uno de los clasificadores para emitir la decisión final [12].

La mayoría de los sistemas que emplean aprendizaje supervisado fueron creados asumiendo que el CE está bien balanceado, es decir, que la representación de patrones por clase es muy pareja. No obstante, esta suposición frecuentemente no es aplicable en el mundo real, ya que existen dominios en los cuales el CE cuenta con desbalance de clases, es decir, una o varias de las clases (minoritarias) está menos representada con respecto al número de patrones pertenecientes a otras clases (mayoritarias).

En la vida real, muchas bases de datos se caracterizan por presentar desbalance en la distribución de los patrones entre las clases, requiriendo especial interés en la correcta clasificación de los patrones contenidos en la(s) clase(s) minoritaria(s), tal es el caso de las bases de datos de información médica. Por ejemplo, en [5] se utiliza una base de datos de mamografías, con 11443 patrones, de los cuales tan solo 260 corresponden a casos de pacientes con micro-calcificaciones en la mama. Al clasificar nuevos casos, la precisión observada es de 97.68% clasificando los nuevos casos como sanos. Esta situación no es nada deseable, ya que muchos de los casos que presentan micro-calcificaciones no fueron detectados como tal.

El problema se complica cuando se trata de CE con más de dos clases, pues resulta difícil determinar qué clases serán consideradas como minoritarias y cuáles como mayoritarias. Se ha comprobado que esta situación puede deteriorar de forma importante la precisión del clasificador, en particular con los patrones que pertenecen a la clase menos representada [2].

La mayoría de las investigaciones que han desarrollado metodologías para lograr el adecuado tratamiento y la disminución de los efectos nocivos que los CE desbalanceados presentan al momento de la clasificación, se enfocan a corregir el desequilibrio de la cantidad de patrones, con tres vertientes básicamente [9]: under-sampling (eliminando patrones) en la clase mayoritaria, over-sampling (replicando patrones) en la clase minoritaria y, por último, internamente

predisponer el proceso de discriminación para compensar el desequilibrio del CE.

En el contexto de redes neuronales, para trabajar con este tipo de problemas se ha propuesto la codificación de las clases, en problemas de dos clases [15]. Es decir, en un CE que dispone de más de dos clases, se codifican las clases con una distribución en dos grupos, de tal modo que en un grupo se incluyen una parte de las clases y en el segundo grupo el resto, esta codificación comúnmente es conocida como *error-correcting output coding* o ECOC.

Este trabajo presenta una exploración de técnicas adaptadas para el tratamiento del desbalance en conjuntos de datos con múltiples clases con la utilización de una ME. Se usa como clasificador individual la regla del vecino más cercano [19]. El artículo tiene la siguiente estructura: En la sección 2 se describen los métodos de ECOC utilizados para el tratamiento del problema de múltiples clases. Los métodos de sub-muestreo utilizados se exponen en la Sección 3, en tanto que los métodos de fusión de decisiones se muestran en la Sección 4. Finalmente, los resultados experimentales son presentados en la Sección 5.

2 Construcción de la mezcla de expertos

Los métodos conocidos como *error-correcting output coding* han sido propuestos para generar diferentes clasificadores que son entrenados como un problema de dos clases en un contexto de múltiples clases [6], [13]. Para ello, las clases representadas en el CE se re-etiquetan temporalmente como pertenecientes a una de esas dos clases. Considerando el siguiente conjunto de etiquetas de clases $\alpha = \{x_1, \dots, x_C\}$, estos métodos codifican en 0's y 1's las etiquetas de clases formando dos grupos, la clase $\alpha(1)$ y la clase $\alpha(0)$. Por ejemplo, si el conjunto de datos tuviera 5 clases $\{x_1, \dots, x_5\}$ y al utilizar algún método para codificar, se obtiene la codificación de las clases de la siguiente manera $[0, 1, 1, 0, 1]$. Cada una de las clases se estaría separando literalmente en dos casos formando dos conjuntos $\alpha(1)$ y $\alpha(0)$. Donde la clases que están codificadas con 0's estarían dentro del

conjunto $\alpha(0)$ y las codificadas en 1's en $\alpha(1)$ respectivamente.

Cada matriz de código puede ser representada como un vector binario, donde el número de espacios estará determinado por el número de clasificadores a construir en la ME. Cada vector estará formado con 1's y 0's y serán diferentes unos de otros (situación conseguida por el método de codificación). El conjunto de todos los vectores formaría la matriz de código. Los métodos utilizados en este estudio se describen a continuación.

2.1 Método uno por clase

Este es el método estándar utilizado para la generación de ECOC. Esta codificación es comúnmente usada, para las salidas de las redes neuronales sobre problemas con múltiples clases. La matriz de código que se obtiene para un problema de 6 clases con este método es la que se muestra en la tabla 1.

Tabla 1. ECOC Uno por clase para $c = 6$ clases ($L = 6$ clasificadores)

	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆
ω_1	0	1	1	1	1	1
ω_2	1	0	1	1	1	1
ω_3	1	1	0	1	1	1
ω_4	1	1	1	0	1	1
ω_5	1	1	1	1	0	1
ω_6	1	1	1	1	1	0

Como se puede observar, con el método se tiene una codificación correspondiente a la matriz identidad, en la que por cada *codeword* se descarta utilizar sólo una clase [13].

2.2 Método exhaustivo

El procedimiento general [8], hace uso de la siguiente ecuación:

$$L=2^{(c-1)}-1 \tag{1}$$

donde L representa el número de clasificadores a construir.

Para garantizar un mejor funcionamiento, se ha sugerido que este método sólo se use, si el número de clases es mayor a tres y menor a siete [13], debido a que cuando se tienen más de 7 clases se tendría una carga computación demasiado grande. Para formar cada codeword se aplican las siguientes reglas:

- La fila 1 está formada por 1's.
- La fila 2 consiste en obtener $2^{(c-2)}$ ceros, los siguientes por $2^{(c-2)}-1$ unos.
- La fila 3 consiste en $2^{(c-3)}$ ceros, los siguientes por $2^{(c-3)}$ unos. Así sucesivamente con todos los clasificadores.
- La siguiente fila i se alternan $2^{(c-i)}$ ceros y unos.
- La última fila es intercalada con unos y ceros.

De este modo, la matriz de código para un conjunto de datos con 4 clases se muestra en la siguiente tabla.

Tabla 2. ECOC obtenido con método exhaustivo con conjunto de datos con 4 clases

	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
ω_1	1	1	1	1	1	1	1
ω_2	0	0	0	0	1	1	1
ω_3	0	0	1	1	0	0	1
ω_4	0	1	0	1	0	1	0

2.3 Método aleatorio

La forma en que trabaja este método, es elegir aleatoriamente qué clases van a pertenecer al conjunto 0 y al conjunto 1, no contempla un número definido de clasificadores a realizar. Para ilustrar el método considere un conjunto $A = (\omega_1, \omega_2, \omega_3, \omega_4)$ de clases. Para construir el clasificador 1, de forma aleatoria se establece que la clase " ω_1 y ω_2 ", forman el grupo 0 y que el resto de las clases en el grupo 1. El clasificador 2 aleatoriamente tendría en el grupo 0 las clases " ω_2 , y ω_3 " y el resto en el grupo 1. Para el clasificador 3 de forma aleatoria tendría en el grupo 1 las clases " ω_3 y ω_1 " y para el grupo 0 las clases " ω_2 y ω_4 ". Para el clasificador 4 tendría en

el grupo 0 las clases " $\omega_1, \omega_2, \omega_3$ " y el resto para el grupo 1 (Tabla 3).

Tabla 3. ECOC aleatorio en un conjunto de datos con 4 clases

	D ₁	D ₂	D ₃	D ₄
ω_1	1	1	1	0
ω_2	1	0	0	0
ω_3	0	0	1	0
ω_4	0	1	0	1

3 Métodos de sub-muestreo

Mediante estos métodos, se realiza la manipulación de patrones para generar múltiples submuestras o subconjuntos, algunos de ellos afectando al tamaño del CE por disminución en el número de patrones resultantes.

Uno de los algoritmos más utilizados para generar subconjuntos es el propuesto por Breiman [4], denominado **Bagging** (Bootstrap Aggregating). Con este algoritmo se generan submuestras llamadas *bootstrap* de tamaño m , construidas mediante la selección aleatoria con reemplazo de patrones contenidos en el CE original, también de tamaño m . En cada ejecución, el algoritmo de aprendizaje utiliza una sub-muestra *bootstrap* diferente. Para cada sub-muestra, cada patrón tiene la probabilidad $1 - (1/m)^m$ de ser seleccionado por lo menos una vez de entre las m veces que se selecciona un patrón. Para valores grandes de m , esto se aproxima a $1 - 1/e \approx 63.2\%$, es decir, cada patrón tiene aproximadamente un 63% de probabilidades de aparecer entre los patrones de entrenamiento de la sub-muestra.

Pese a la obtención de una adecuada codificación de las clases, esto no garantiza que al momento de realizar la distribución de las clases en los grupos se elimine o disminuya el desbalance existente entre las clases. Es posible que algunos grupos manejen el desbalance en su interior [16]. Por esta situación es fundamental atender este efecto no deseado del tratamiento de desbalance, de lo contrario, el desempeño del clasificador podría verse seriamente afectado.

Para atender este problema, en esta investigación al método de sub-muestreo se le ha realizado una modificación. Se establece un umbral h (eq. 2) que determine si una clase se considera mayoritaria o minoritaria.

$$h = \frac{m}{c} \quad (2)$$

donde m = total de patrones y c es el número de clases.

Con la determinación del umbral, las clases cuyo número de patrones es mayoritario al umbral, son consideradas como clases mayoritarias.

Una vez identificadas la categoría de las clases (mayoritarias o minoritarias), de las consideradas mayoritarias se seleccionan tantos patrones como el valor del umbral, en tanto que de las consideradas como minoritarias, se toma el total de patrones que contienen.

4. Fusión de clasificadores

Uno de los más populares métodos para obtener la decisión final en una ME, es la fusión de las decisiones. La fusión asume que todos los clasificadores son competitivos y complementarios (igualmente *expertos*). Por este motivo, cada uno de ellos emite una decisión respecto a cada patrón de prueba que se presenta. La salida de los clasificadores es un vector H-dimensional que contiene las decisiones de cada uno de los H clasificadores:

$$[D_{i,1}(\mathbf{y}), \dots, D_{i,H}(\mathbf{y})]^T \quad (3)$$

Del conjunto de salidas, se toma la decisión final mediante la aplicación y/o utilización de alguno de los siguientes principios: votación por mayoría simple o votación por mayoría ponderada.

4.1 Votación por mayoría no ponderada

Dentro de esta investigación se aplicó el método más tradicional y sencillo; la votación por mayoría simple. Básicamente este método

consiste en proporcionar un voto con valor de 1 a la clase que predijo el clasificador, al final se suman los votos de cada una de las clases y se determina como la clase ganadora aquella que tiene el número mayor de votos [10].

4.2 Votación por mayoría ponderada

Para asignar el peso w a los clasificadores individuales, se implementaron dos métodos: considerando la distribución (previa al proceso de sub-muestreo) de patrones por clase y considerando el error en la clasificación de patrones por clase. En lo que respecta al primer método, se implementó la siguiente ecuación:

$$w = \frac{m_{c_i}}{m} \quad (4)$$

donde m_{c_i} es el número patrones de la clase i y m es el total de patrones en el CE.

Una vez aplicado el método, se logró garantizar que a los patrones de las clases minoritarias tengan mayor reconocimiento en el proceso de clasificación.

5. Experimentación

Para la experimentación realizada se utilizaron 4 conjuntos de datos reales obtenidos del Repositorio de la Universidad de California (UCI, <http://archive.ics.uci.edu/ml/>) cuyas características principales se muestran en la Tabla 4. De cada CD, se estimó su precisión general mediante el método de validación cruzada con la obtención de 5 muestras, utilizando un 80% de los patrones para entrenamiento y el 20% restante para conjunto de prueba.

Tabla 4. Conjuntos de datos

CD	Clase						
	1	2	3	4	5	6	7
Vehicle	218	212	217	199			
Winne	59	71	84				
Segment							
Glass	70	76	17	13	9	29	

5.1 Métodos para análisis de resultados

Para evaluar la efectividad del sistema se utiliza una matriz de confusión como la mostrada en la Tabla 5. Los elementos en esta tabla caracterizan el desempeño en términos de precisión de un sistema dado.

De la Tabla 5 es posible obtener la precisión general (ecuación 6), medida ampliamente utilizada para el análisis de resultados en reconocimiento de patrones.

$$Precisión\ general = \frac{\sum_{i=1}^m n_{ii}}{N} \tag{6}$$

donde N es el número total de patrones y $n_{ii}=n_{i+}$ es la *Precisión por clases*.

Tabla 5. Matriz de confusión para un problema de múltiples clases

Clase proporcionada por el clasificador	Clases reales				Total (n _{i+})
	1	2	...	m	
1	n ₁₁	n ₁₂	...	n _{1m}	n ₁₊
2	n ₂₁	n ₂₂	...	n _{2m}	n ₂₊
...
m	n _{m1}	n _{m2}	...	n _{mm}	n _{m+}
Total (n _{+j})	n ₊₁	n ₊₂	...	n _{+m}	N

Muchos estudios han establecido que la precisión general no es el mejor criterio para medir el desempeño del clasificador en dominios de desbalance.

Por ejemplo, considere un dominio donde sólo el 5% de los patrones corresponden a la clase minoritaria. En este caso, si se clasifican todos los nuevos patrones como pertenecientes a la clase mayoritaria, la precisión general proporcionaría un índice de 95%. Como es posible ver, este resultado se aleja de lo real, pues al desconocer los patrones de la clase menos representada se estaría omitiendo identificar patrones de gran importancia para el problema de estudio.

Para evitar esta situación, es necesario utilizar otro criterio que refleje con más certeza el

rendimiento del clasificador en contextos de desbalance. Para fines de este estudio se utiliza el coeficiente Kappa.

Basado en la matriz de confusión de la Tabla 2 es factible obtener el coeficiente y la varianza del Kappa. La idea fundamental implica el análisis de las diferencias entre los datos de referencia y los datos entrantes determinados por la diagonal principal.

Kappa (ecuación 7) es un indicador que adquiere valores entre 0 y 1, representando el primero la absoluta falta de concordancia y el segundo, concordancia total. Un valor menor a 0.4 se considera como expresión de concordancia insuficiente entre 0.41-0.60 como moderada y mayor que 0.60 elevada. Este método puede medir la exactitud de manera más precisa que la matriz de confusión por calcular, no solamente los valores de sus columnas de los extremos, sino también los contenidos en el interior de la matriz [14].

$$\hat{K} = \frac{m \sum_{i=1}^{\gamma} x_{i,i} - \sum_{i=1}^{\gamma} (x_{i+} * x_{+i})}{m^2 - \sum_{i=1}^{\gamma} (x_{i+} * x_{+i})} \tag{7}$$

donde γ es el número de filas, $x_{i,j}$ número de patrones en una fila i y una columna j , x_{i+} y x_{+i} totales marginales de una fila i y una columna i respectivamente, y m número de patrones evaluados.

Cuando el resultado obtenido por el Kappa muestra que la concordancia es pobre, esto puede indicar mayor confiabilidad, a diferencia de cuando se obtiene concordancia elevada. Esto se debe a que, si en la muestra estudiada la proporción de una de las alternativas de calificación es mucho más frecuente que la otra (por ejemplo, gran disparidad entre la proporción de individuos con y sin daño) y para ella la concordancia es adecuada, el resultado expresará una elevada concordancia global, pero en realidad, está informando la concordancia para la alternativa de calificación de mayor frecuencia. Por ello, se ha propuesto calcular por separado la concordancia para cada alternativa, permitiendo una visión más equilibrada del comportamiento del método.

5.2 Resultados

Para la integración de la ME se consideró la manipulación de los patrones utilizando los métodos ECOC. Para fines de clasificación, se utilizó la regla del vecino más cercano aplicada a cada uno de los miembros del ME, combinando sus salidas con los esquemas de votación simple y votación ponderada. Los resultados obtenidos con el clasificador único (sin combinación) han sido incluidos como valores de referencia. La Tabla 6 muestra los resultados de la clasificación expresados tanto en precisión general, como coeficiente Kappa.

Como puede apreciarse, los resultados de la Tabla 6 respecto a precisión general (PG) y coeficiente Kappa muestran diferencias significativas respecto al desempeño del clasificador con cada conjunto de datos. Estas diferencias están dadas por los aspectos descritos en la sección 5.1.

Tabla 6. Resultados de clasificación utilizando el conjunto de datos original

	Kappa	P.G
Glass	58.5	70.0
Vehicle	52.2	99.5
Segment	95.6	97.1
Wine	55.4	67.7

Para determinar las diferencias significativas de los resultados, entre cada uno de los métodos analizados, se calculan los *rank* de cada método [17]. En este caso, para cada conjunto de datos, el método con la mayor precisión recibe rango 1, y el peor recibe un *rank* de 6. Si hay un empate, el resultado es compartido entre los métodos empatados. Para cada método, hay 4 valores de clasificación, uno para cada conjunto, así, el rango general de un método es el rango promedio de este método a través de los 4 conjuntos de datos.

Para completar el análisis de significancia estadística entre los resultados obtenidos, se utilizó la prueba de Nemenyi [17]. La prueba obtiene una diferencia crítica (DC) para rechazar

las hipótesis en las que el valor de p correspondiente es menor que el ajustado α , la ecuación utilizada es la siguiente [18]:

$$DC = q_{\alpha} \sqrt{\frac{K(K+1)}{5N}} \quad (8)$$

donde, q_{α} es el valor crítico, K es el número de métodos a comparar y N es el número de conjunto de entrenamiento utilizado. Para este cálculo, q_{α} se basa en el estadístico del rango estudentizado dividida por $\sqrt{2}$, esto puede observarse en la Tabla 7 [17].

Tabla 7. Valores críticos para la prueba de Nemenyi

Clasif.	2	3	4	5	6	7
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693

Por lo expuesto previamente y debido a que en este artículo se están comparando 3 diferentes métodos para aplicar los ECOC con dos métodos de fusión; el valor correspondiente de $q_{0.05}$ es 2.85 y para $q_{0.10}$ es de 2.59.

Los resultados experimentales son proporcionados en la Tabla 8, donde se incluyen los resultados obtenidos en la fusión ponderada y no ponderada de clasificadores. La tabla se encuentra dividida en dos partes, en la primera de ellas se muestran los resultados de la clasificación expresados tanto en precisión general (PG), como con el coeficiente Kappa.

En la segunda parte (línea inferior de la tabla) se proporcionan los valores del *rank* general para cada uno de los métodos utilizados

Como puede observarse para el CD Wine, por la naturaleza del método exhaustivo, no fue posible construir la mezcla de expertos.

De los resultados mostrados en la tabla 7 se pueden observar varias situaciones. Primeramente, al analizar los tres métodos de codificación, encontramos que la mejor representación y distribución de patrones es obtenida con el uno por clase, esto es confirmado

Tabla 8. Coeficiente Kappa y precisión general obtenidos con los diferentes métodos de fusión, los valores en negrita enfatizan los mejores resultados de cada problema

CD	Uno por clase				Aleatorio				Exhaustivo			
	Ponderada		Simple		Ponderada		Simple		Ponderada		Simple	
	Kappa	P.G	Kappa	P.G	Kappa	P.G	Kappa	P.G	Kappa	P.G	Kappa	P.G
Glass	45.7	58.0	50.0	62.5	40.0	51.5	44.4	57.5	25.7	35.5	44.4	57.5
Segment	95.5	96.2	95.5	96.2	89.9	96.3	89.9	96.4	48.4	70.0	48.4	70.0
Vehicle	47.1	62.1	47.4	48.3	46.4	57.8	48.8	60.6	44.0	57.9	50.4	60.9
Wine	51.0	68.8	54.0	70.0	47.1	69.4	51.7	69.4	-----	-----	-----	-----
Rank general	2.6	2.1	1.6	2.4	4.4	3.8	2.8	2.9	5.8	5.2	8.2	3.4

por los *rank* generales obtenidos (2.6 y 1.6, según Kappa). De igual modo, se logra identificar que pese a lo elaborado del método exhaustivo, es el que en promedio proporciona los peores resultados con los CD utilizados.

Por otro lado, analizando el comportamiento de los métodos de fusión utilizados, la votación por mayoría simple ofrece mejores resultados en casi todos los casos y cuando no es así, proporciona resultados muy similares a los de la fusión ponderada.

El gráfico de la Figura 1, muestra las diferencias obtenidas en los valores críticos entre los métodos de ECOC respecto al *rank* general.

Para interpretar los resultados mostrados en la Figura 1, se establece que un determinado método A es significativamente diferente al B, si el *rank* general (A)+DC < *rank* general de (B). En este sentido, observando las diferencias

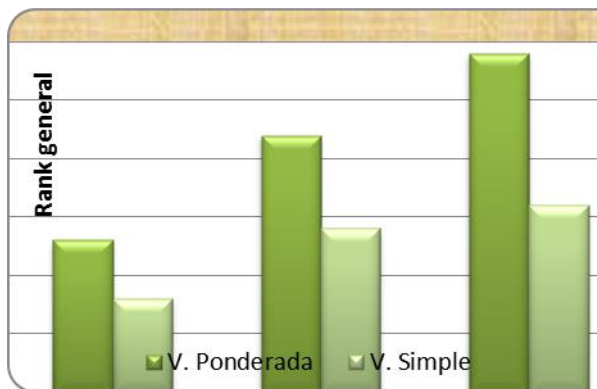


Fig. 1. Rank general obtenido por el coeficiente Kappa

significativas entre el menor *rank* obtenido (1.6) y el mayor (5.8), se puede concluir que el rendimiento del método uno por clase con votación simple muestra una mejora estadísticamente significativa respecto a los otros dos métodos.

6. Conclusiones

En este estudio, se presenta la adaptación de modelos utilizados en contextos de redes neuronales (ECOC), para realizar la codificación de las clases en contextos donde el conjunto de datos cuenta con más de dos clases utilizando el modelo de ME, de forma adicional, se propone establecer un umbral para que sea utilizado por el método de sub-muestreo con la intención de disminuir el desbalance que pudiera existir en los grupos formados por las codificaciones obtenidas.

Para construir las submuestras que integrarían la Mezcla de Expertos, se adecuaron algunos métodos de sub-muestreo ampliamente estudiados en la literatura. La adecuación consistió en la implementación de un umbral basado en la cardinalidad de patrones por clase de tal manera que se guió la selección de patrones para evitar el efecto no deseado del desbalance en las submuestras resultantes. Una vez obtenidas las submuestras y las codificaciones se construyó la Mezcla de Expertos en la que se incorporó una medida de ponderación estática con la intención de incrementar la eficiencia del modelo.

Con la utilización del coeficiente Kappa y la prueba de significancia estadística de Nemenyi, fue posible realizar un análisis más apegado a la naturaleza de los conjuntos de datos utilizados y obtener las siguientes conclusiones: pudo mostrarse la conveniencia de utilizar la ME para tratar de forma más eficiente el proceso de clasificación al utilizar múltiples clases, el métodos de codificación que favorece una mejor representación y distribución de patrones es el método uno por clase. De igual modo, es posible identificar que la utilización del método exhaustivo muestra bajo rendimiento. Esta situación es conveniente si se observa que el método uno por clase es el que requiere menos costo computacional y tiene mayor simplicidad en su construcción (ver sección 2.1).

Respecto a los métodos de fusión, no se logra observar diferencias significativas, por lo que, con los datos aquí utilizados no se identifica mayor ventaja la utilización de un método u otro.

Por último, realizar la adaptación de los métodos de codificación ECOC para el tratamiento de múltiples clases ha sido la primera aproximación de su uso, si bien, es una idea atractiva, al construir ME, observando los índices de precisión mostrados, es evidente la necesidad de profundizar en su estudio mediante la utilización de métodos de bajo (limpieza y condensado) y alto muestreo con la intención de mejorar el rendimiento que en los momentos los métodos ofrecen. De igual modo, se contempla la implementación de nuevos métodos de ponderación (basados en distancias) que ayuden a mejorar el rendimiento del clasificador. Por último, el estudio comparativo con otros métodos utilizados para el tratamiento del desbalance en múltiples clases, principalmente en el contexto de redes neuronales es un estudio en proceso de realización.

Agradecimientos

Este trabajo fue realizado gracias al apoyo recibido de los proyectos: 3072/2011 de la UAEM, PROMEP/103.5/12/4783 de las SEP, SDMAIA-010 del TESjo, UR-001 del ININ.

Referencias

1. **Alejo, R., Sotoca, J.M., & Casañ, G.A. (2008).** An empirical study for the multi-class imbalance problem with neural networks. *Pattern Recognition, Image Analysis and Applications. Lecture Notes in Computer Science*, 5197, 479–486.
2. **Barandela, R., Sánchez, J.S., García, V., & Rangel, E. (2001).** Fusion of techniques for handling the imbalanced training sample problem. *6th Ibero-American Symposium on Pattern Recognition, Florianópolis, Brazil*, 34–40.
3. **Batista, G., Patri, R.C., & Monard, M.C. (2004).** A study of the Behavior of several methods for Balancing Machine Learning Training data. *ACM SIGKDD Explorations Newsletter*, 6(1), 20–29.
4. **Breiman, L. (1998).** Arcing classifiers. *The Annals of Statistics*. 26(3), 801–849.
5. **Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2000).** SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1), 321–357.
6. **Dietterich, T.G. & Bakiri, G. (1994).** Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2(1), 263–286.
7. **Dietterich, T.G. (1997).** Machine learning research: four current directions. *AI Magazine*, 18(4), 97–136.
8. **Kong, E.B. & Dietterich, T.G. (1995).** Error-Correcting Output Coding Corrects Bias and Variance. *12th International Conference on Machine Learning*. California, USA, 313–321.
9. **Eavis, T. & Japkowicz, N. (2000).** A recognition-based alternative to discrimination-base multilayer perceptrons. *Advances in Artificial Intelligence, Lecture Notes in Computer Science*, 1822, 280–292.
10. **Kuncheva, L.I. (2000).** Clustering-and-selection model for classifier combination. *4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES'2000)*. Brighton, UK, 1, 185–188.
11. **Kuncheva, L.I. (2001).** Using measures of similarity and inclusion of multiple classifier fusion by decision templates. *Fuzzy Sets and Systems*, 122(3), 401–407.
12. **Kuncheva, L.I. & Whitaker, C.J. (2003).** Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2), 181–207.
13. **Kuncheva, L.I. (2005).** Using diversity measures for generating error-correcting output codes in

classifier ensemble. *Pattern Recognition Letters*, 26, 83–90.

14. **Congalton, R.G. & Green, K. (1999).** *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. Boca Raton: Lewis Publications.
15. **Daqi, G., Wei, W., & Jianliang, G. (2007).** Class-modular multi-layer perceptions, task decomposition and virtually balanced training subsets. *International Joint Conference on Neural Networks*, Orlando, Florida, USA, 2153–2158.
16. **Soda, P. & Iannello, G. (2010).** Decomposition Methods and Learning approaches for Imbalanced Dataset: An Experimental Integration. *20th International Conference on Pattern Recognition*, Istanbul, Turkey, 3117–3120.
17. **Demsar, J. (2006).** Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(2006), 1–30.
18. **García, S. & Herrera, F. (2008).** An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research*, 9(12), 2677–2694.
19. **Dasarathy, B.V. (1991).** *Nearest Neighbor (NN) Norms: nn Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press.



Rosa María Valdovinos Rosas, Doctora en Ciencias Computacionales. Los intereses de investigación se centran en el estudio de Reconocimiento de Patrones y Minería de datos con redes neuronales, algoritmos

genéticos y métodos estadísticos no paramétricos.



Rosalinda Abad Sánchez, Maestra en Ciencias de la Computación, las líneas de investigación se orientan al tratamiento de complejidades de datos, como el desbalance en dos y múltiples clases.



Roberto Alejo Eleuterio, Doctor en Sistemas Informáticos Avanzados. Su área de investigación son las redes neuronales artificiales aplicadas al reconocimiento de patrones.



Edgar Herrera Arriaga, Doctor en Ciencias Nucleares. Los intereses de **Investigación** se centran en métodos estadísticos y probabilísticos.



Adrian Trueba Espinosa, Doctor en Ciencias. Líneas de investigación. Bases de datos heterogéneas, reconocimiento de patrones, Sistemas de Información Geográfica

Article received on 05/11/2012, accepted on 21/06/2013.