Selección de características distribuida en entornos heterogéneos

Verónica Bolón-Canedo
Grupo LIDIA. DCITIC.
Universidade da Coruña
A Coruña, España
veronica.bolon@udc.es

Rubén Seoane-Martínez
Grupo LIDIA. CITIC.
Universidade da Coruña
A Coruña, España

José Luis Morillo-Salas
Grupo LIDIA. CITIC.
Universidade da Coruña
A Coruña, España
jose.luis.morillo@udc.es

Amparo Alonso-Betanzos
Grupo LIDIA. CITIC.
Universidade da Coruña
A Coruña, España
ciamparo@udc.es

***Resumen*—Los avances en las Tecnologías de la Información y las Comunicaciones han contribuido a la proliferación de grandes bases de datos. En algunos casos estos datos ya están distribuidos en su origen, pero en otros casos su gran escala hace que el procesamiento en un único nodo sea imposible, y en consecuencia la distribución en varios nodos de cómputo es una opción natural para su manejo. En este trabajo, proponemos una metodología que nos permite distribuir el proceso de selección de características, la mayoría de las veces un paso de preprocesado imprescindible en los conjuntos de alta dimensión actuales, ya que nos permite reducir la dimensión de entrada, seleccionando las características relevantes y eliminando las redundantes y/o irrelevantes. En particular, nuestra propuesta en este artículo se centra en el problema de los conjuntos de datos desbalanceados, bien porque la situación se da ya en origen o bien cuando este contexto en que las distintas clases de datos no están igualmente representadas en las distintas particiones se produce debido a que se debe distribuir el conjunto único original para poder tratarlo. Los resultados experimentales obtenidos demuestran que nuestra aproximación distribuida obtiene resultados de error comparables a la aproximación centralizada, aportando como ventajas una reducción apreciable del tiempo computacional y la capacidad de trabajar eficientemente en entornos de desbalanceo de clases.**

***Index Terms*—selección de características, algoritmos distribuidos, conjuntos de datos desbalanceados.**

I. Introducción

La selección de características (SC) es una técnica de aprendizaje automático en la que se seleccionan los atributos que permiten que un problema esté claramente definido, mientras que los irrelevantes o redundantes se ignoran [1]. Tradicionalmente, un algoritmo de SC se aplica de manera centralizada, es decir, se utiliza un único modelo selector de características sobre todos los datos del conjunto para resolver un problema determinado. Sin embargo, en algunos casos, los datos pueden o bien estar ya distribuidos en varias localizaciones, o bien se puede usar una estrategia de aprendizaje distribuido para repartir en varios nodos de cómputo un conjunto de datos que es demasiado grande para poder ser procesado en un único nodo. De esta forma podemos aprovechar el procesamiento de estos múltiples subconjuntos de datos bien en secuencia o en paralelo. Existen varias formas de distribuir una tarea de selección de características, aunque las más comunes son:

- los datos están juntos en un conjunto de datos muy grande, por lo que se distribuyen en varios procesadores, se ejecuta un algoritmo de SC idéntico en cada uno y luego los resultados parciales se combinan para obtener un resultado final, y
- los datos pueden estar en diferentes conjuntos de datos situados en diferentes ubicaciones, por lo que se ejecuta un algoritmo de selección de características idéntico en cada uno y los resultados se combinan para obtener un resultado final.

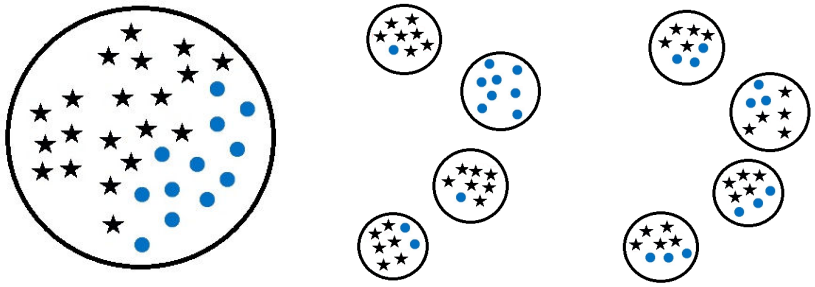
Al respecto, existen varios trabajos en la literatura que realizan la selección de características de forma distribuida [2], [3]. Sin embargo, cuando los datos se distribuyen en varios procesadores, pueden aparecer algunos problemas adicionales, como un alto desequilibrio entre clases en algunos de los nodos, o incluso la situación extrema en la que algunas clases no están representadas en absoluto en algunos de los subconjuntos de datos. El *problema de desequilibrio de clase o desbalanceo* se produce cuando un conjunto de datos está dominado por una clase mayoritaria que tiene significativamente muchas más instancias que las otras clases, llamadas minoritarias. En este caso, los algoritmos de aprendizaje computacional suelen presentar un sesgo hacia las clases mayoritarias, ya que las reglas que predicen correctamente esas instancias se ponderan positivamente a favor de la métrica de precisión, mientras que las reglas específicas que predicen ejemplos de la clase minoritaria generalmente se ignoran. Por lo tanto, las muestras de las clases minoritarias se clasifican erróneamente más a menudo que las de las otras clases [4].º

En este trabajo presentamos una metodología para distribuir el proceso de SC, que tiene en cuenta este problema de la posible heterogeneidad de los subconjuntos. Para ello usamos dos alternativas: (i) forzar las particiones del conjunto de datos para mantener el equilibrio entre las clases, y (ii) aplicar técnicas de sobremuestreo (oversampling) cuando el desequilibrio es inevitable.

II. Metodología distribuida

En este trabajo, se detalla la aplicación de una metodología para distribuir el proceso de SC sobre la base de trabajos pre-

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (proyectos de investigación TIN 2015-65069-C2-1-R y la Red Española de Big Data y Análisis de datos escalable, TIN2016-82013-REDT), y por Fondos de Desarrollo Regional de la Unión Europea.



(a) Centralizado (b) Partición aleatoria (c) Partición homogénea

Figura 1. Escenarios centralizado (a) y particiones aleatoria (b) y homogénea (c) en un proceso de selección de características distribuido

vios [5], [6]. Esta metodología consta de tres pasos principales, que son los siguientes:

1. partición de los datos, si éstos no estuviesen ya distribuidos en origen,
2. aplicación del método de SC a cada una de las diferentes particiones realizadas
3. combinación de los resultados.

Debemos de tener en cuenta que los dos primeros pasos se repiten varias rondas (r), para garantizar la captura de suficiente información para el paso de combinación de los resultados parciales.

El primer paso de la metodología anterior es el núcleo de este trabajo y consiste en dividir sin reemplazo los datos del conjunto original, asignando grupos de n muestras a cada subconjunto de datos. Se seguirán dos enfoques principales: *partición aleatoria*, en la que se realizará una distribución aleatoria de los datos en los distintos nodos, y *partición homogénea*, en la que se mantienen las proporciones del conjunto original en cada uno de los subconjuntos obtenidos. Un ejemplo de estos dos tipos de partición, junto con el escenario centralizado en el que todos los datos están juntos, se puede ver en la figura 1.

Después de realizar una partición, el conjunto de datos podría estar desequilibrado (ya sea porque la partición se realizó al azar o porque el conjunto de datos ya estaba desbalanceado en origen). En este caso, nuestra propuesta consiste en aplicar el método de sobremuestreo SMOTE [7], que agrega ejemplos sintéticos de la clase minoritaria al conjunto de datos original hasta que la distribución de clases se equilibre. Para poder conseguir esto, SMOTE genera ejemplos sintéticos de la clase minoritaria utilizando los ejemplos originales de la misma de la siguiente manera: en primer lugar, busca los k vecinos más cercanos de la muestra de la clase minoritaria que se utilizará como base para la nueva muestra sintética. Luego, en el segmento que une la muestra de la clase minoritaria con uno o todos sus vecinos, se toma aleatoriamente una muestra sintética y se agrega al nuevo conjunto de datos sobremuestreados.

El siguiente paso en la metodología general consiste en aplicar un método de SC en cada partición. Las características que se seleccionan para ser eliminadas reciben un voto y luego, se realiza una nueva ronda que conduce a una nueva partición del conjunto de datos y se lleva a cabo una nueva iteración de la votación hasta alcanzar el número predefinido de rondas r. Finalmente, las características que han recibido una cantidad de votos por encima de un umbral predefinido se eliminan. Por lo tanto, se obtiene finalmente un conjunto único de características que se pueden utilizar para entrenar un clasificador C y probar su rendimiento en un nuevo conjunto de muestras (conjunto de datos de test). Más detalles sobre cómo elegir el umbral de votos se pueden encontrar en [5], [6]. El pseudocódigo de la metodología propuesta se muestra en el algoritmo 1.

```
1  inicializar el vector de votos a 0
2  para cada ronda hacer
3      dividir el conjunto de datos d aleatoriamente o
         mantener las proporciones de las clases en
         subconjuntos de datos disjuntos
4      para cada subconjunto de datos hacer
5          si los datos están desbalanceados entonces
6              aplicar SMOTE
           fin
7          aplicar un algoritmo de selección de características
8          incrementar un voto para cada característica a ser
             eliminada
       fin
   fin
9  eliminar las características cuyo número de votos sea
     superior a un umbral
10 clasificar con el subconjunto de características obtenido
```

Algoritmo 1: Pseudo-código de la metodología propuesta

III. Resultados experimentales

En esta sección presentaremos el esquema de experimentación y los resultados obtenidos. Recordemos que los objetivos de la experimentación son dos, (i) poder establecer qué tipo de partición es la más adecuada y (ii) cúal es la influencia del algoritmo de sobremuestreo SMOTE cuando las particiones presentan datos desbalanceados.

III-A. Comparación entre las aproximaciones distribuidas y la centralizada

Con este objetivo, hemos seleccionado 6 conjuntos de datos, cuyas características se resumen en la Tabla I, y que están

Cuadro I
Características de los conjuntos utilizados en la primera parte de la experimentación, en donde solamente se utiliza SMOTE en la clase minoritaria.

Conjunto	Nº Muestras	Nº Características	Nº Clases	% clase mayoritaria
Connect4	67557	42	3	65.83
Isolet	7797	617	26	3.85
Madelon	2400	500	2	50
Mnist	60000	717	2	50
Ozone	2536	72	2	97.12
Spambase	4601	57	2	60.6

disponibles para su descarga en el UCI Machine Learning Repository [1].

Aunque la metodología propuesta es genérica, y por lo tanto se puede usar con cualquier método de SC, en este trabajo hemos elegido una suite de cuatro filtros, basados en diferentes tipos de métricas. Concretamente hemos utilizado Correlation-Based Feature Selection (CFS), Consistency-based, Information Gain y ReliefF, todos ellos disponibles en la herramienta de software libre Weka [2]. Para posteriormente poder evaluar los resultados de la selección de características realizada, hemos elegido cuatro clasificadores populares en el estado del arte: C4.5, Naive Bayes, IB1 y Vectores de Máquinas Soporte (en inglés, Support Vector Machine –SVM–). Los experimentos se realizaron en una CPU Core ™i3-6100 Intel ®3.70 GHz con 16 GB de memoria RAM.

En el primer estudio experimental se compararon tres escenarios diferentes: (i) la aproximación centralizada estándard, (ii) la distribución aleatoria, y (iii) el particionado homogéneo. Para las dos aproximaciones distribuidas (la aleatoria y la homogénea), el número de rondas utilizado fue de 5. Para asegurar una buena fiabilidad en los resultados obtenidos, se realizó una validación *hold-out* estándard, es decir, se dividieron los distintos conjuntos de la tabla I en dos subconjuntos diferentes, con la proporción 2/3 para entrenamiento y 1/3 para prueba, y se repitió esta operación 5 veces. También se han usado test de significación estadística, en primer lugar un test de Friedman para comprobar si existían diferencias significativas para un nivel de significación $\alpha = 0{,}5$, y posteriormente de Nemenyi para obtener aquellos modelos que no son significativamente diferentes a los que obtienen la mayor precisión. Las tablas detalladas con los resultados obtenidos para todas las combinaciones entre conjuntos de datos, métodos de SC y clasificadores pueden verse en el material suplementario que se encuentra en [3].

La Tabla II muestra un resumen de este primer conjunto de experimentos, en los que la meta es comparar los tres escenarios (centralizado, partición aleatoria y partición homogénea), independientemente de la aplicación de la técnica de *oversampling* SMOTE. La tabla muestra los resultados para cada combinación de conjunto de datos y escenario, teniendo en cuenta dos medidas de evaluación diferentes: la precisión de la clasificación y el valor del índice kappa. El motivo de incluir el valor de Kappa es porque éste evalúa la calidad del aprendizaje teniendo en cuenta las situaciones en las que el conjunto de datos está desequilibrado y el clasificador aprende correctamente la clase mayoritaria, pero sistemáticamente clasifica erróneamente las instancias de la clase minoritaria. En las primeras dos filas de cada conjunto de datos, se puede consultar el promedio de la precisión de clasificación y Kappa; y en las últimas dos filas se muestran los valores máximos de precisión y Kappa (y la combinación que lo obtiene entre paréntesis). Como se puede ver, los enfoques distribuidos (partición aleatoria y homogénea) son una buena solución para disminuir el tiempo computacional sin implicar una degradación en el rendimiento de clasificación. Comparando los dos enfoques distribuidos, vale la pena señalar que, en general, el enfoque homogéneo parece obtener resultados más estables, mientras que con la partición aleatoria puede ocurrir que en un caso particular la proporción de las clases sea óptima y por esa razón obtiene los mejores resultados en algunos casos.

Como era de esperar, los enfoques distribuidos reducen significativamente el tiempo de ejecución en comparación con el enfoque centralizado (ver detalles en el material complementario 3^0), aunque depende concretamente tanto del método de selección empleado como del conjunto de datos. Cuando el conjunto de datos es pequeño, la mejora es leve (por ejemplo, de 0.40s a 0.34s en el conjunto Ozone) pero en conjuntos de datos más grandes, la mejora es considerable (por ejemplo, de 820.46s a 0.83s en el conjunto Connect-4). Es remarcable también el buen rendimiento obtenido por los métodos de selección ReliefF y Consistency-based.

III-B. Utilización de SMOTE en las particiones

El segundo grupo de experimentos consiste en la evaluación de la efectividad de SMOTE ante el problema del desbalanceo de clases. La comparación se realizó entre los dos escenarios distribuidos (partición aleatoria y homogénea). Debemos tener en cuenta que, al aplicar la partición aleatoria, es posible que algunos subconjuntos de datos estén desbalanceados, incluso si el conjunto de datos completo no lo estaba. Por lo tanto, hemos aplicado SMOTE cuando el subconjunto de datos no estaba balanceado (ya sea bien debido a la existencia de esta circunstancia en clase original, o bien debido a la partición aleatoria). Se han realizado diferentes experimentos con diferentes porcentajes de sobremuestreo. Por ejemplo, si la clase minoritaria tiene 40 muestras y aplicamos SMOTE

[1] http://archive.ics.uci.edu/ml/index.php
[2] http://www.cs.waikato.ac.nz/ ml/weka/
[3] http://lidiagroup.org/index.php/en/materials-en.html

Conjunto		Centralizado	Aleatorio	Homogéneo
Connect4	Precisión (media)	65.72	67.52	67.52
	Kappa (media)	0.175	0.184	0.188
	Precisión (max)	73.37 (Cons+C4.5)	74.12 (Rel+C4.5)	72.81 (IG+C4.5)
	Kappa (max)	0.454 (Cons+C4.5)	0.425 (Rel+C4.5)	0.399 (Rel+C4.5)
Isolet	Precisión (media)	63.92	69.12	68.64
	Kappa (media)	0.624	0.678	0.673
	Precisión (max)	84.60 (Rel+SVM)	85.51 (Rel+SVM)	83.87 (Rel+SVM)
	Kappa (max)	0.839 (Rel+SVM)	0.849 (Rel+SVM)	0.832 (Rel+SVM)
Madelon	Accuracy (media)	74.64	72.25	75.32
	Kappa (media)	0.492	0.444	0.506
	Accuracy (max)	88.75 (Varios+IB1)	81.75 (Rel+C4.5)	89.62 (Rel+IB1)
	Kappa (max)	0.774 (Varios+IB1)	0.636 (Rel+C4.5)	0.792 (Rel.+IB1)
MNIST	Precisión (media)	81.04	83.68	83.83
	Kappa (media)	0.620	0.672	0.675
	Precisión (max)	89.96 (Rel+IB1)	96.33 (Cons+IB1)	95.83 (Rel+IB1)
	Kappa (max)	0.799 (Rel+IB1)	0.926 (Cons+IB1)	0.916 (Rel+IB1)
Ozone	Precisión (media)	92.09	91.06	90.87
	Kappa (media)	0.1014	0.1012	0.092
	Precisión (max)	97.12 (Todos+SVM)	96.99 (Todos+SVM)	96.97 (Todos+SVM)
	Kappa (max)	0.189 (Cons+C4.5)	0.215 (Cons+C4.5)	0.180 (IG+IB1)
Spambase	Precisión (media)	86.66	87.18	87.63
	Kappa (media)	0.723	0.732	0.742
	Precisión (max)	91.42 (CFS+C4.5)	91.24 (CFS+C4.5)	91.73 (CFS+C4.5)
	Kappa (max)	0.819 (CFS+C4.5)	0.816 (CFS+C4.5)	0.826 (CFS+C4.5)

Cuadro II

Resumen de los resultados obtenidos para las aproximaciones distribuidas y centralizada. No se ha utilizado el método SMOTE en las aproximaciones distribuidas.

con un nivel de 100, significa que se generan 40 muestras sintéticas, si el nivel es 200, significa que se generan 80 nuevas muestras. Además, incluimos la opción "auto", que consiste en aplicar SMOTE de tal forma que las clases queden completamente balanceadas.

Conjunto	Precisión	Kappa	Escenario	Combinación	SMOTE
Isolet	85.68	0.851	Aleatorio	Rel+SVM	Auto
Madelon	89.63	0.792	Homogéneo	Rel+IB1	0
MNIST	96.34	0.926	Aleatorio	Cons+IB1	0
Connect4	74.12	0.425	Aleatorio	Rel+C4.5	0
	72.90	0.442	Aleatorio	Rel+C4.5	100
Ozone	97.28	0	Homogéneo	All+SVM	100
	90.82	0.302	Homogéneo	Rel+SVM	600
Spambase	91.73	0.826	Homogéneo	CFS+C4.5	0
	91.68	0.827	Homogéneo	CFS+C4.5	300

Cuadro III

Resumen de los resultados obtenidos usando SMOTE en las clases minoritarias.

```
1  para repeticiones hacer
2      dividir el conjunto de datos d aleatoriamente en
         subconjuntos disjuntos de datos train y test
3      calcular la clase mayoritaria del conjunto de train
4      para cada nivel Smote minoritaria hacer
5          para cada nivel Smote mayoritaria hacer
6              train = SMOTE(clase_mayoritaria, train)
7              train = SMOTE(clase_minoritaria, train)
8              para filtros hacer
9                  train_filtrado = sel_car(filtro,train)
10                 para clasificador hacer
11                     clasificar(clasificador, train_filtrado,
                         test)
                   fin
               fin
           fin
       fin
   fin
```

Algoritmo 2: Pseudo-código de la metodología propuesta usando SMOTE también en la clase mayoritaria

La tabla III muestra el resumen de los mejores resultados obtenidos al aplicar diferentes niveles de sobremuestreo con SMOTE a los subconjuntos de datos. En la primera fila de cada conjunto de datos, se muestra la opción con la mayor precisión, mientras que la segunda fila representa la opción con el valor Kappa más alto. Cuando el mejor resultado para ambas mediciones de evaluación se logra mediante la misma combinación y escenario, solo se muestra una fila. Como era de esperar, la aplicación de una técnica de sobremuestreo no es necesaria en el caso de conjuntos de datos equilibrados (Isolet, Madelon, MNIST). En el caso de Isolet, la aplicación de SMOTE en el escenario de partición aleatoria ha resultado provechosa, ya que en este caso es posible que los conjuntos de datos equilibrados produzcan subconjuntos de datos no balanceados (especialmente para Isolet, con un alto número de clases). Por el contrario, los conjuntos de datos desbalanceados (Connect4, Ozone y Spambase) son buenos candidatos para mejorar sus resultados después de aplicar SMOTE. De hecho, la Tabla III muestra que la aplicación del método de sobremuestreo mejora los valores Kappa, lo que significa que el aprendizaje de las clases es mejor. Hay que recordar que la clase mayoritaria de ozono tiene el 97.12 % de las muestras, por lo que al obtener una precisión de clasificación del 97.28 % es posible que clasifique correctamente todas las muestras de la clase mayoritaria, pero solo unas pocas de la clase minoritaria. Después de aplicar el método de sobremuestreo, la precisión cae al 90.82 %, probablemente porque el clasificador no está tan sobreajustado para aprender la clase mayoritaria

y tiene una tasa de verdaderos positivos más alta en la clase minoritaria.

Finalmente, se realizó un tercer conjunto de experimentos, en los que se utiliza también la técnica SMOTE para añadir también muestras sintéticas en la clase mayoritaria, no sólo en la minoritaria, de forma que todas las clases, mayoritarias y minoritarias, cuenten en sus subconjuntos con muestras sintéticas, como se puede ver en el algoritmo 2.

Para realizar este tercer bloque de experimentos se utilizaron dos tipos de conjuntos, los que denominamos con la etiqueta *normal* en la tabla IV, que son conjuntos de datos en los que el número de muestras es mucho mayor que el número de características, y conjuntos de datos del tipo *Microarrays* [8], obtenidos de investigaciones sobre la clasificación de casos de cáncer, que tienen un elevado número de características y un número muy pequeño de muestras (ver tabla IV). La idea es comprobar no sólo si el realizar SMOTE en todas las clases mejora el resultado, al tener muestras sintéticas en todas ellas, sino también si el balance entre muestras y características influye en los resultados. Se han repetido de nuevo los experimentos, pero en este caso además se han añadido muestras sintéticas también en la clase mayoritaria, utilizando SMOTE con porcentajes de 20, 40 y 100. Al igual que anteriormente, se han obtenido valores para todas las posibles combinaciones de clasificador, filtro y combinación de porcentajes de SMOTE en la clase mayoritaria. En la tabla V se pueden ver los resultados obtenidos para todos los conjuntos de la tabla IV sin SMOTE, con la alternativa de SMOTE en la clase minoritaria y con la alternativa de usar SMOTE en todas las clases.

Como podemos ver en la tabla V, la alternativa SMOTE en las clases minoritaria y mayoritaria conjuntamente es siempre la opción que alcanza la precisión máxima, con los valores de índice kappa más altos (en ocasiones, otras alternativas consiguen idénticas kappas, y la alternativa SMOTE sólo en minoritaria empata en precisión máxima en 5 de los 12 conjuntos). No parecen existir grandes diferencias entre los dos tipos de conjuntos, si bien la diferencia en precisión media entre las dos alternativas usando SMOTE en los conjuntos microarray es menor que en el caso de los conjuntos que hemos denominado normales.

IV. Conclusiones y Trabajo Futuro

Hemos presentado una metodología para la selección de características distribuida, tratando de resolver el problema del desbalanceo de los datos en las diferentes particiones. Para ello, hemos forzado a las particiones de datos en los diferentes nodos a mantener la misma distribución de clase que el conjunto de datos original y hemos aplicado la técnica de sobremuestreo (oversampling) SMOTE. Los resultados experimentales en siete conocidos conjuntos de datos han demostrado que:

- El enfoque distribuido — partición aleatoria u homogénea — es competitivo cuando se compara con el enfoque centralizado estándar, incluso en algunos casos mejorando el rendimiento de clasificación.
- La partición homogénea obtiene resultados más estables que la partición aleatoria.
- La aplicación de SMOTE en las clases minoritarias (uso estándard del procedimiento), mejora la calidad del aprendizaje en conjuntos de datos desbalanceados, en algunos casos a expensas de una ligera disminución en la precisión general.
- Además al aplicar el método propuesto con un porcentaje pequeño de SMOTE también en la clase mayoritaria se aprecia una mejora en la precisión máxima obtenida en todos los conjuntos de datos. Además, si bien es cierto que esta última alternativa no obtiene prácticamente en ningún caso los mejores valores de precisiones medias, sí que consigue obtener los valores de kappa más altos, por lo que el método presenta una mayor robustez.

Como trabajo futuro, nos planteamos probar otros métodos para tratar la heterogeneidad, como puede ser el caso de las técnicas de submuestreo (undersampling en inglés), ponderación, etc.

Referencias

[1] I. Guyon. *Feature extraction: foundations and applications*, volume 207. Springer, 2006.

[2] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos. Centralized vs. distributed feature selection methods based on data complexity measures. *Knowledge-Based Systems*, 117:27–45, 2017.

[3] V. Bolón-Canedo, N. Sánchez-Maroño, and Alonso-Betanzos. Distributed feature selection: An application to microarray data classification. *Applied Soft Computing*, 30:136–150, 2015.

[4] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[5] V. Bolón-Canedo, N. Sánchez-Maroño, and J. Cerviño-Rabuñal. Scaling up feature selection: a distributed filter approach. In *Conference of the Spanish Association for Artificial Intelligence*, pages 121–130. Springer, 2013.

[6] V. Bolón-Canedo, N. Sánchez-Marono, and J. Cervino-Rabunal. Toward parallel feature selection from vertically partitioned data. In *Proceedings of ESANN 2014, pp. 395–400*, 2014.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, 2011.

[8] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J.M. Benítez, and F. Herrera. A review of microarray datasets and applied feature selection methods. *Information Sciences*, 282:111–135, 2014.

Cuadro IV
Características de los conjuntos del tercer bloque de experimentación, con muestras sintéticas en la clase mayoritaria

Conjunto	Tipo	Nº Muestras	Nº Características	Nº Clases	% clase mayoritaria
Arrhythmia	Normal	452	279	16	54.2
Connect4	Normal	67557	42	3	65.83
Musk2	Normal	6598	168	2	63
Nomao	Normal	34465	120	2	71.44
Ozone	Normal	2536	72	2	97.12
Spambase	Normal	4601	57	2	60.6
Weight	Normal	4024	90	5	34.04
Brain	Microarray	21	12625	2	67
CNS	Microarray	60	7129	2	75
Colon	Microarray	62	2000	2	65
Gli85	Microarray	85	22283	2	69
Ovarian	Microarray	253	15154	2	64

Conjunto		Sin SMOTE	Solo SMOTE minoritaria	SMOTE minoritaria + mayoritaria
Arrhythmia	Precisión (media)	**63.07**	62.63	62.07
	Kappa (media)	0.739	0.746	**0.774**
	Precisión (max)	68.34 (CFS+Naive)	67.68 (CFS+Naive+100)	**68.74** (Rel+SVM+300+40)
	Kappa (max)	**1** (Todos+IB1)	**1** (Todos+IB1+Todos)	**1** (Todos+IB1+Todos+Todos)
Connect4	Precisión (media)	66.14	**66.20**	62.15
	Kappa (media)	0.163	0.430	**0.478**
	Precisión (max)	76.12 (Cons+C4.5)	77.88 (Cons+C4.5+600)	**78.21** (Cons+C4.5+100+20)
	Kappa (max)	0.777 (Cons+C4.5)	**1** (Cons+IB1+Todos)	**1** (Cons+IB1+Todos+Todos)
Musk2	Precisión (media)	**89.41**	87.35	85.17
	Kappa (media)	0.672	0.687	**0.739**
	Precisión (max)	95.62 (Cons+C4.5)	95.44 (CFS+C4.5+100)	**95.72** (Cons+C4.5+100+40)
	Kappa (max)	**1** (Todos+IB1)	**1** (Varios+IB1+Varios)	**1** (Varios+IB1+Todos+Todos)
Nomao	Precisión (media)	**85.75**	84.22	83.94
	Kappa (media)	0.646	0.693	**0.723**
	Precisión (max)	94.34 (Cons+C4.5)	94.52 (Cons+C4.5+300)	**94.70** (Cons+C4.5+Auto+100)
	Kappa (max)	0.964 (Cons+C4.5)	**1** (Cons+IB1+Todos)	**1** (Cons+IB1+Todos+Todos)
Ozone	Precisión (media)	**92.36**	91.88	88.58
	Kappa (media)	0.271	0.506	**0.628**
	Precisión (max)	96.99 (Cons+Todos)	**97.02** (Rel+SVM+600)	**97.02** (Rel+SVM+300+Varios)
	Kappa (max)	0.982 (Info+Rel)	**1** (CFS+IB1+Todos)	**1** (CFS+IB1+Todos+Todos)
Spambase	Precisión (media)	85.92	**87.45**	87.11
	Kappa (media)	0.800	0.858	**0.859**
	Precisión (max)	92.27 (CFS+C4.5)	92.79 (CFS+C4.5+300)	**92.85** (CFS+C4.5+Auto+100)
	Kappa (max)	0.998 (Cons+IB1)	**1** (Cons+IB1+Auto)	**1** (Cons+IB1+Varios+Varios)
Weight	Precisión (media)	84.40	85.44	**86.33**
	Kappa (media)	0.803	0.829	**0.851**
	Precisión (max)	**100** (Cons+Varios)	**100** (Cons+Varios+Todos)	**100** (Varios+C4.5+Todos+Todos)
	Kappa (max)	**1** (Cons+Varios)	**1** (Varios+IB1+Varios)	**1** (Varios+IB1+Varios+Varios)
Brain	Precisión (media)	59.11	**62.27**	62.14
	Kappa (media)	0.882	0.889	**0.904**
	Precisión (max)	82.86 (Info+C4.5)	**94.29** (CFS+C4.5+Todos)	**94.29** (CFS+C4.5+Todos+Todos)
	Kappa (max)	**1** (CFS+Todos, Info+Todos)	**1** (CFS+Todos, Info+Todos)	1 (CFS+Todos, Info+Todos)
CNS	Precisión (media)	55.38	**58.98**	58.13
	Kappa (media)	0.867	0.910	**0.921**
	Precisión (max)	65 (Cons+SVM)	**70** (CFS+C4.5+600)	**70** (CFS+Naive+300+20)
	Kappa (max)	**1** (Todos+IB1)	**1** (Todos+IB1+Todos)	1 (Todos+IB1+Todos+Todos)
Colon	Precisión (media)	**77.62**	77.17	76.81
	Kappa (media)	0.861	0.913	**0.920**
	Precisión (max)	86.67 (Naive+Rel)	87.62 (Info+Naive+Auto)	**88.67** (Rel+Naive+Auto+20)
	Kappa (max)	1 (Todos+IB1)	**1** (Todos+IB1+Todos)	**1** (Todos+IB1+Todos+Todos)
Gli85	Precisión (media)	77.99	**80.40**	80.08
	Kappa (media)	0.944	0.969	**0.976**
	Precisión (max)	85.71 (Info+SVM)	88.57 (Cons+IB1+Varios)	**90** (CFS+IB1+Auto+100)
	Kappa (max)	**1** (Todos+IB1)	**1** (Todos+IB1+Todos)	**1** (Todos+IB1+Todos+Todos)
Ovarian	Precisión (media)	97.68	98.04	**98.09**
	Kappa (media)	0.989	0.994	**0.995**
	Precisión (max)	**100** (CFS+SVM)	**100** (CFS+SVM+Todos)	**100** (CFS+Varios+Varios)
	Kappa (max)	**1** (CFS+SVM, Todos+IB1)	**1** (CFS+SVM+Todos, Todos+IB1+Todos)	**1** (Todos+IB1+Todos+Todos)

Cuadro V
Resumen de los resultados obtenidos utilizando las tres aproximaciones, sin utilizar SMOTE, usando SMOTE sólo en la clase minoritaria–utilización estándard– y usando SMOTE en todas las clases.