Selección de Patrones de Entrenamiento Representativos en Muestras Desequilibradas mediante Voto Mayoritario en Perceptrones Paralelos*

Iván Cantador y José R. Dorronsoro
Dept. de Ingeniería Informática e Instituto de Ingeniería del Conocimiento
Universidad Autónoma de Madrid, 28049 Madrid

Resumen

En problemas de clasificación reales es común que existan patrones que no aportan información relevante a la construcción de los discriminantes, al estar representados por otros mucho más significativos, y patrones que la dificultan, por tener ciertas componentes ruidosas. Esto complica el aprendizaje de clasificadores, que puede bien centrarse exclusivamente en la clase mayoritaria o bien no ser capaz de evitar la influencia de los ejemplos ruidosos. Con el fin de reducir la complejidad de los datos y proporcionar un mayor grado de generalización, puede resultar beneficioso trabajar sólo con los ejemplos más representativos para definir las fronteras de separación. Como una aproximación a la identificación y selección de estos patrones, se propone una estrategia que emplea el voto mayoritario de los novedosos Perceptrones Paralelos y el concepto de margen de activación que surge en su entrenamiento, y que considera el desequilibrio entre clases otorgando cierta preferencia a los patrones minoritarios para su selección. Estas ideas se aplicarán al entrenamiento de los bien conocidos Perceptrones Multicapa, y se comprobará de forma empírica en varios problemas de clasificación una importante reducción de las muestras de entrenamiento, a la vez que una mejora de la generalización en los clasificadores obtenidos con ellas.

1. Introducción

De gran interés son los problemas de clasificación difíciles originados en el mundo real que pueden agruparse bajo el epígrafe de "Muestras Desequilibradas", y que se caracterizan por tener clases de interés cuyo número de ejemplos es mucho menor que el de otras, que oscurecen o dificultan la identificación de las primeras, o bien presentar un alto número de patrones ruidosos, cuyas características son propias de una clase, pero cuyo etiquetado corresponde a otra. Ejemplos de esta situación se dan en problemas tan diversos como la detección de fraude en transacciones con tarjetas de crédito [4] y en llamadas telefónicas [7], la gestión de las comunicaciones [6], el diagnóstico médico de enfermedades no usuales [9], o la categorización de textos [8].

Estas circunstancias dificultan la construcción de clasificadores, que pueden bien centrarse exclusivamente en la clase mayoritaria o bien no ser capaces de evitar la influencia de los patrones ruidosos. Prácticamente cualquier clasificador es sensible a este problema, pero su influencia es mucho mayor en aquellos de mayor capacidad aproximante, como puede ser el caso de los Perceptrones Multicapa (PMCs), donde el problema se apreció desde su primer uso.

De hecho, si bien la comunidad científica que trabaja en Aprendizaje Automático asumía que la distribución de clases natural es la mejor para el aprendizaje, esta asunción fue descartada al comprobarse el efecto negativo que el desequilibrio existente entre el número de muestras minoritarias y mayoritarias puede provocar en la eficacia de los clasificadores [14]. El problema fue entonces abordado mediante dos tipos de enfoques generales: el remuestreo de los conjuntos de entrenamiento para equilibrar el número de representantes de cada clase [3][15], y la asignación de costes a las muestras de tal modo

* Con apoyo parcial del proyecto TIC 01-572, TIN 2004-07676.

que en el proceso de entrenamiento se premie la correcta clasificación de los ejemplos con mayores costes [13].

Dentro del primer enfoque se pueden distinguir a su vez dos técnicas diferentes: la eliminación o *submuestreo* de ejemplos de entrenamiento mayoritarios, y la replicación o *sobremuestreo* de ejemplos de entrenamiento minoritarios. Aunque siendo favorables en general, ambas poseen inconvenientes. Mientras que la primera tiene el riesgo de que se pierda información relevante, la segunda incrementa el tamaño del conjunto de entrenamiento, y por tanto el tiempo necesario para el aprendizaje, además de poder dar lugar a sobreajuste, por hacer uso de réplicas de ejemplos.

Como alternativa, se han propuesto estrategias basadas en seleccionar para el entrenamiento sólo aquellos ejemplos que están cerca de la frontera de clasificación [5][10], y siguiendo esta idea, en este trabajo se presenta una aproximación que emplea los novedosos Perceptrones Paralelos (PP) presentados por Auer et al. en [1] y que busca la identificación y eliminación de aquellos patrones de entrenamiento que son: (1) "redundantes", al no aportar información relevante, pues están bien representados por otros ejemplos, o que son (2) de "etiquetado ruidoso" (ruidosos a partir de ahora), al tener atributos que indican al clasificador la pertenencia a una clase incorrecta, entorpeciendo de este modo el aprendizaje.

Experimentalmente se comprobará que la técnica propuesta reduce considerablemente los conjuntos de entrenamiento y mejora las capacidades de generalización de clasificadores obtenidos con las nuevas muestras. Para ello se trabajará con PMCs como máquina de aprendizaje final e independiente al proceso de filtrado de patrones, tanto por su ya mencionado alto poder computacional y el riesgo de sobreajuste (pérdida de generalización) que este último conlleva, como por ser una de las herramientas más utilizadas para construcción de clasificadores.

Una vez introducidas en la sección 2 la arquitectura y la regla de aprendizaje de los PPs, las secciones 3 y 4 describirán el procedimiento seguido para equilibrar y seleccionar los ejemplos de entrenamiento representativos. La evaluación empírica de este último se realizará en las secciones 5 y 6, mientras que las conclusiones y posibles líneas de investigación futura se plantearán en la sección 7.

2. Perceptrones Paralelos

Los Perceptrones Paralelos tienen una estructura análoga a las de las bien conocidas máquinas comité [12]. Formados por un número impar de perceptrones estándar P_h con predicciones ± 1, su salida unidimensional es simplemente el signo de la suma de esas predicciones, i.e., el voto mayoritario de las mismas. Son, de este modo, adecuados para problemas de discriminación de 2 clases, aunque, como se muestra en [1], también pueden emplearse en problemas de regresión.

En más detalle, supóngase que se trabaja con patrones D-dimensionales $\mathbf{X} = (X_1,\ldots, X_D)^t$, donde la D-ésima componente se fija a 1 para considerar efectos de sesgo. Si la máquina comité (MC) tiene H perceptrones, cada uno de los cuales con vector de pesos $\mathbf{W}_h$, para una entrada $\mathbf{X}$ la salida del perceptrón h es $P_h(\mathbf{X}) = s(\mathbf{W}_h \cdot \mathbf{X}) = s(\mathrm{act}_h(\mathbf{X}))$, con $s(\cdot)$ denotando la función signo y $\mathrm{act}_h(\mathbf{X}) = \mathbf{W}_h \cdot \mathbf{X}$ la activación del perceptrón h debida a $\mathbf{X}$.

Teniendo entonces que

$$\sum_{h=1}^{H} P_h(\mathbf{X}) = N_+(\mathbf{X}) - N_-(\mathbf{X}) = N(\mathbf{X})$$

donde $N_\pm$ representa el número de salidas positivas/negativas de los perceptrones, la salida final $f(\mathbf{X})$ de la MC es $f(\mathbf{X}) = s(N(\mathbf{X}))$, cogiendo H impar para evitar empates. Asumiendo que cada entrada $\mathbf{X}$ tiene asociada una etiqueta $y_\mathbf{X} = \pm 1$, la salida $f(\mathbf{X})$ es correcta si $y_\mathbf{X} \cdot f(\mathbf{X}) = +1$.

Si no es el caso, i.e. si $y_\mathbf{X} \cdot f(\mathbf{X}) = -1$, el entrenamiento de las MC clásicas ([12], capítulo 6) intenta cambiar el menor número de salidas de perceptrones de tal modo que $\mathbf{X}$ pueda ser correctamente clasificado, eligiendo aquellos perceptrones incorrectos para los cuales $|\mathrm{act}_h(\mathbf{X})|$ es más pequeño, y cambiando sus pesos mediante la bien conocida regla de Rosenblatt:

$$\mathbf{W}_h = \mathbf{W}_h + \eta \cdot y_\mathbf{X} \cdot \mathbf{X} \qquad (1)$$

Sin embargo, en el entrenamiento de Perceptrones Paralelos la actualización (1) se aplica a todos los perceptrones incorrectos, esto es, a aquellos P_h que verifican que $y_\mathbf{X} \cdot P_h(\mathbf{X}) = -1$.

Adicionalmente, cuando un patrón $\mathbf{X}$ es clasificado correctamente, el entrenamiento de un PP también aplica un procedimiento de estabilización mediante la ecuación (1) en aquellos perceptrones con $0 < y_\mathbf{X} \cdot \mathrm{act}_h(\mathbf{X}) < \gamma$, para

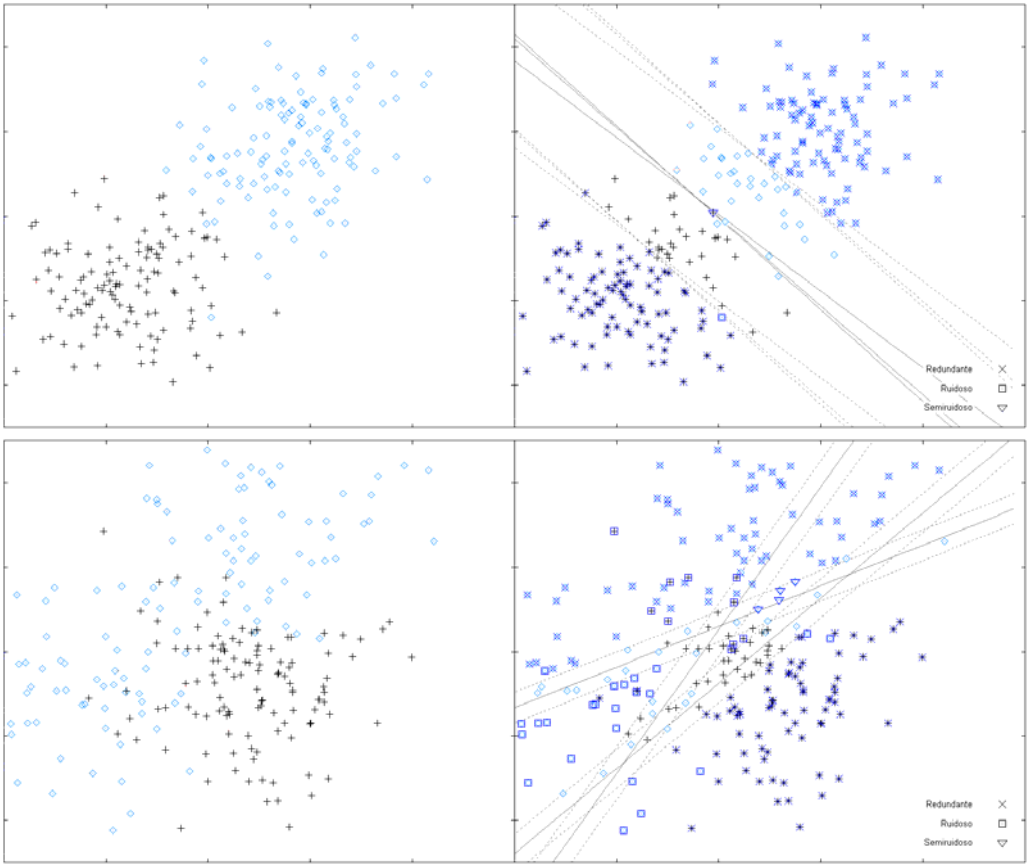

Figura 1. Ejemplos de identificación de patrones redundantes, ruidosos y semiruidosos en *twonorm* y *threenorm*.

un cierto margen $\gamma > 0$. Nótese que para ellos una pequeña perturbación podría causar una asignación de clase errónea, y es conveniente alejar de 0 el valor de sus activaciones $\text{act}_h(\mathbf{X})$.

El margen γ se ajusta dinámicamente. Concretamente, como se propone en [1], y siguiendo un entrenamiento *batch*, después de haber presentado todos los patrones en una época de aprendizaje, el valor de γ se incrementa a $\gamma + 0.25\eta$ por cada patrón $\mathbf{X}$ que cumpla en todos los perceptrones que $y_\mathbf{X} \cdot \text{act}_h(\mathbf{X}) > \gamma$, mientras que se decrementa a $\gamma - 0.75\eta$ por cada $\mathbf{X}$ que cumpla en al menos un perceptrón que $0 < y_\mathbf{X} \cdot \text{act}_h(\mathbf{X}) < \gamma$.

Además, para asegurar la convergencia del margen, la tasa de aprendizaje η se actualiza con la regla $\eta_t = \eta_0 / \sqrt{t}$, donde η_0 es el valor inicial de la tasa y t es la época de entrenamiento actual.

3. Equilibrado de muestras

La estabilización de las salidas del PP a través del margen γ puede usarse para decidir qué patrones tienen sus proyecciones en el espacio de atributos mejor situadas respecto a las fronteras de separación alcanzadas.

Como se introdujo en la sección anterior, un patrón $\mathbf{X}$ está bien clasificado por un perceptrón h si cumple que $y_\mathbf{X} \cdot \text{act}_h(\mathbf{X}) = y_\mathbf{X} \cdot \mathbf{W}_h \cdot \mathbf{X} > 0$. La regla de aprendizaje modifica los vectores de pesos $\mathbf{W}_h$ de los perceptrones que clasifican correctamente

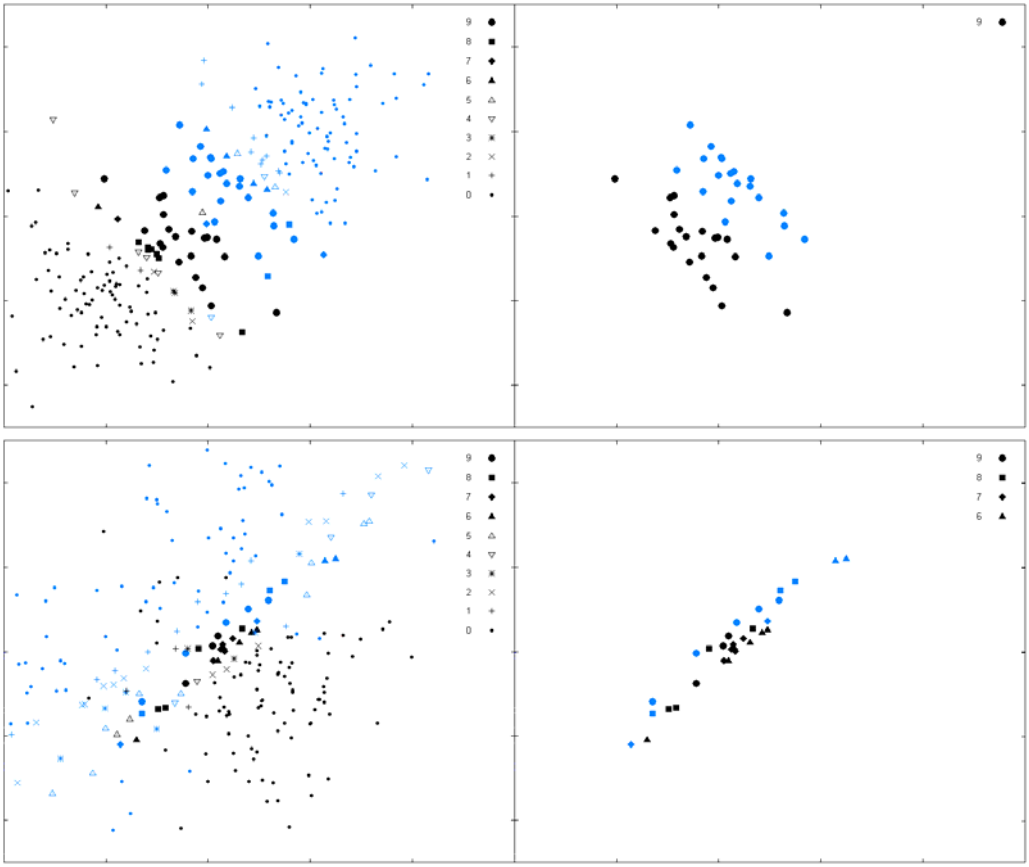

Figura 2. Conjuntos de entrenamiento originales y filtrados con M=9 y M=6, en los problemas *twonorm* y *threenorm*.

al patrón pero con un margen demasiado pequeño, i.e., con $0 < y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) < \gamma$. Por esta razón, los ejemplos que por el contrario poseen una activación normalizada $y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) > \gamma$ son buenos candidatos de ser considerados "seguros", en el sentido de que pequeñas modificaciones de los pesos no alterarán la clase que el PP les asigne.

De forma análoga, los patrones $\mathbf{X}$ con $y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) < 0$ están mal clasificados por el perceptrón h, y si cumplen que $y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) < -\gamma$ serán buenos candidatos de ser considerados "de etiquetado erróneo", pues sólo modificaciones relativamente grandes de los pesos harán cambiar el valor de la clase asignada.

Con estas ideas, a la hora de definir las fronteras de separación, ni los patrones "seguros", ni los "de etiquetado erróneo" son útiles. Por ello, la estrategia de equilibrado de muestras propuesta considera estas categorías, pero en todos los perceptrones que forman un PP. De este modo, para un PP dado se establece que un patrón $\mathbf{X}$ es *redundante* si la mayoría de los perceptrones cumplen que $y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) > \gamma$, y que es *ruidoso* si la mayoría de los perceptrones cumplen que $y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) < -\gamma$.

De manera más formal, para un PP de H perceptrones se establece que un patrón $\mathbf{X}$ es:

- **Redundante** si $\#\{h: y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) > \gamma\} > \lfloor H/2 \rfloor$.
- **Ruidoso** si $\#\{h: y_{\mathbf{X}} \cdot \text{act}_h(\mathbf{X}) < -\gamma\} > \lfloor H/2 \rfloor$.

Los patrones redundantes no aportan información relevante, pues están alejados de las

fronteras, mientras que los ejemplos ruidosos entorpecen el aprendizaje, pues obligan a cambiar los pesos, haciendo probable que se modifique la etiqueta asignada a otros patrones antes clasificados correctamente. La selección de patrones de entrenamiento descartará entonces los patrones redundantes y ruidosos, y mantendrá el resto, que, al tener sus activaciones con valores cercanos a 0, serán los de proyecciones más cercanas a las fronteras de separación.

Por otra parte, de manera experimental se ha comprobado que es beneficioso eliminar también aquellos patrones que están clasificados incorrectamente por todos los perceptrones de un PP, independientemente de que sus activaciones caigan dentro o fuera del margen γ. En muestras desequilibradas, sin embargo, es peligroso eliminar todos estos patrones "semiruidosos", pues se incrementa el riesgo de dejar la clase minoritaria sin representantes. Por esta razón, de ellos sólo se descartarán los que pertenezcan a la clase mayoritaria.

Se define entonces un patrón **representativo** como aquel que no es redundante, ruidoso o semiruidoso de la clase mayoritaria. Como ejemplo ilustrativo, la Figura 1 muestra la identificación de cada una de las categorías de patrones definidas sobre dos sencillos problemas de clasificación en 2 dimensiones: *twonorm* y *threenorm* [2], en los que respectivamente se busca la separación de 2 y 3 nubes de patrones siguiendo distribuciones normales. En ella, se puede observar que con un PP de 3 perceptrones se identifican con gran precisión muchos patrones no representativos (redundantes y ruidosos) y algunos pocos considerados semiruidosos, adecuados para ser eliminados.

4. Selección de Patrones Representativos

Expuesto en la sección anterior el criterio de reducción y equilibrado de muestras a través de un PP, la estrategia de selección de patrones representativos es sencilla, pues consiste simplemente en combinar las categorizaciones realizadas por *T* Perceptrones Paralelos diferentes.

De forma iterativa, se aplica el siguiente proceso selectivo. Inicialmente ningún patrón del conjunto de entrenamiento S_{tr} es candidato a ser representativo: para cada uno de ellos se inicializa a 0 un contador que irá guardando el número de veces que han sido definidos como relevantes por los diversos PPs. Posteriormente, se generan *T* Perceptrones Paralelos diferentes que darán su opinión acerca de la influencia de todos los patrones y que incrementarán en 1 los contadores de aquellos que consideren representativos. Con el fin de asegurar mayor diversidad en las fronteras de separación obtenidas, los PPs se construyen mediante selección con reemplazamiento (*bootstrap*) del conjunto de entrenamiento original. Finalmente, aquellos patrones cuyo contador tenga un valor igual o superior a un cierto valor *M*, i.e. han sido considerados representativos por al menos *M* de los PPs, se mantendrán en conjunto de entrenamiento final S_f. El resto serán descartados.

Tras el filtrado, a partir de S_f se genera un clasificador independiente que será evaluado en un conjunto de test S_{te} no usado en el aprendizaje. El Algoritmo 1 resume el mecanismo de filtrado explicado e incluye al Perceptrón Multicapa (PMC) como método de aprendizaje final, ajeno al proceso de selección.

```
Entrada:
  Str: Conjunto de entrenamiento
  Ste: Conjunto de test
  M: Umbral de la selección

Para X ∈ Str hacer
  Contador[X] = 0;
Fin
Para t = 1,…,T hacer
  St ← bootstrap(Str);
  PPt ← construirPP(St);
  Para X ∈ Str hacer
    Si (esRepresentativo(X,PPt))
     Contador[X] = Contador[X]+1;
  Fin
Fin
Sf = {};
Para X ∈ Str hacer
  Si (Contador[X] ≥ M)
   Sf ← Sf ∪ {X};
Fin

Salida:
   PMCf ← construirPMC(Sf);
   accf ← evaluarPMC(PMCf,Ste);
```

Algoritmo 1. Esquema general de la selección de patrones representativos y evaluación de la misma con un PMC.

Problema	Patrones				Atributos	
	# total	# may	# min	Distribución	# continuos	# discretos
pima	768	500	268	34.9	8	0
breast-w	699	458	241	34.5	9	0
vowel	990	900	90	9.1	10	0
sick	3772	3541	231	6.1	6	21
abalone	731	689	42	5.7	7	1

Tabla 1. Descripción de los conjuntos de datos empleados en los experimentos.

De nuevo se hace uso de representaciones gráficas para aclarar las ideas expuestas. La Figura 2 incluye la identificación y selección de patrones representativos en los problemas *twonorm* y *threenorm* para distintos valores de *M*. Se puede observar que el mecanismo proporciona una importante reducción de los conjuntos de entrenamiento, manteniendo los patrones más cercanos a las fronteras de clasificación.

5. Experimentos

Para la evaluación empírica de la estrategia de selección de patrones de entrenamiento mediante Perceptrones Paralelos se han elegido 5 problemas de la UCI [11] con diferentes grados de desequilibrio entre clases, oscilando del 34.9% al 5.7% de ejemplos minoritarios. La Tabla 1 resume las características de todos ellos, que se han colocado en orden creciente según nivel de desequilibrio. En los problemas no binarios (*vowel*, *abalone*) los experimentos se han realizado clasificando los ejemplos de la clase minoritaria contra el resto de patrones, reunidos en una segunda clase.

En el filtrado de los conjuntos de datos se han empleado PPs con 3 perceptrones. Como se demuestra en [1], esta arquitectura es capaz de resolver satisfactoriamente problemas de clasificación comunes de la literatura. Los valores iniciales de la tasa de aprendizaje y del margen se fijaron respectivamente en $\eta = 0.01$ y $\gamma = 0.05$. El número de épocas de entrenamiento para cada PP se estableció en 250, tiempo que se comprobó suficiente para la convergencia de los parámetros de los clasificadores. El número de PPs creados para la selección fue de $T = 10$. Como se verá en la siguiente sección, se han probado diferentes valores del umbral *M*. Por su parte, la evaluación de la estrategia se ha realizado con un PMC de una única capa oculta de 3 unidades, suficiente para alcanzar porcentajes de error similares a los reportados en la literatura [11][14], y entrenado mediante descenso por gradiente estocástico con una tasa de aprendizaje constante $\eta = 0.01$ durante 10000 épocas, bastante para la convergencia del error de entrenamiento. Tanto en los PPs como en los PMCs los vectores de pesos finales elegidos fueron los de la época en la que se produjo el menor error de entrenamiento.

6. Resultados

En general, una de las medidas más empleadas en la evaluación de algoritmos de aprendizaje es, sin duda, el porcentaje de patrones de test clasificados correctamente o *accuracy* (*acc*). Sin embargo, al abordar problemas desequilibrados esta opción puede ser contraproducente, pues es necesario que se haga una predicción satisfactoria tanto en la clase minoritaria, como en la mayoritaria. Por ello, se empleará la media geométrica *g* [10] definida como

$$g = \sqrt{acc^{+} \cdot acc^{-}} \qquad (2)$$

donde acc^{+} y acc^{-} son respectivamente los porcentajes de aciertos en la clasificación de patrones de test de la clase minoritaria (positiva) y mayoritaria (negativa). El valor de *g* refleja un compromiso con la predicción de ambas clases.

En la Tabla 2 se reúnen los resultados experimentales obtenidos. De izquierda a derecha, las columnas contienen: el nombre del problema considerado, el parámetro *M* empleado en la selección de patrones ($M = 0$ indica aprendizaje sobre el conjunto de entrenamiento inicial), los

Problema	*M*	*acc*	acc^+	acc^-	*g*	# pat	# may	# min	$\|S_f\|$ relativo	Distr.
pima	0	74.091	59.978	81.640	70.0	614	400	214	100.0	34.9
	1	74.740	63.112	80.960	**71.5**	401	226	175	**65.3**	43.6
	2	73.698	60.720	80.640	***70.0***	309	167	141	***50.3***	45.8
	3	70.106	59.251	77.280	67.7	235	123	112	38.2	47.7
	4	67.581	58.714	72.320	65.2	171	86	85	27.9	49.9
	5	65.420	61.219	67.680	64.4	108	51	57	17.7	53.0
breast-w	0	96.108	95.604	96.376	96.0	559	366	193	100.0	34.5
	1	96.680	97.508	96.244	96.9	338	267	71	60.5	21.1
	2	96.766	97.922	96.156	97.0	271	207	64	48.6	23.7
	3	96.766	98.087	96.067	97.1	222	162	60	39.8	27.1
	4	96.852	98.337	96.067	***97.2***	196	140	56	***35.1***	28.5
	5	96.794	98.087	96.111	**97.1**	178	125	53	**31.8**	29.7
vowel	0	99.636	98.667	99.734	99.2	792	720	72	100.0	9.1
	1	99.737	99.556	99.756	**99.7**	375	303	72	**47.4**	19.1
	2	99.273	98.667	99.333	***99.0***	275	205	71	***34.8***	25.6
	3	98.647	96.889	98.822	97.9	214	147	67	27.1	31.4
	4	98.546	94.667	98.933	96.8	175	112	64	22.1	36.4
	5	97.576	93.334	98.000	95.6	152	92	61	19.2	39.8
sick	0	98.648	92.466	99.151	95.8	3017	2790	227	100.0	7.5
	1	98.696	92.462	99.203	95.8	2310	2115	195	76.6	8.4
	2	98.457	94.289	98.796	***96.5***	1966	1782	184	***65.2***	9.4
	3	98.240	94.088	98.578	**96.3**	1515	1341	174	**50.2**	11.5
	4	98.128	92.242	98.607	95.4	1175	1015	160	39.0	13.6
	5	97.577	92.253	98.011	95.1	883	738	146	29.3	16.5
abalone	0	94.501	41.611	96.605	63.4	585	551	34	100.0	5.8
	1	89.986	47.278	92.605	66.2	344	315	29	58.8	8.5
	2	92.400	46.389	96.344	**66.9**	295	267	28	**50.4**	9.4
	3	91.278	46.111	94.048	***65.9***	255	229	26	***43.6***	10.4
	4	87.906	47.422	90.389	65.5	238	212	26	40.7	10.9
	5	88.704	45.444	91.356	64.4	224	198	26	38.2	11.6

Tabla 2. Resumen de resultados experimentales obtenidos. Los mejores, según compromiso entre *g* y $|S_f|$, en negrita.

porcentajes de aciertos globales (*acc*), de la clase minoritaria (acc^+) y mayoritaria (acc^-), los valores de *g*, los números de patrones totales (pat), mayoritarios (may) y minoritarios (min), así como el tamaño relativo de los conjuntos de entrenamiento y la distribución de estos descrita como el porcentaje de patrones minoritarios.

Los valores expuestos se calcularon mediante validación cruzada en 5 particiones, y fueron promediados en 5 iteraciones. Las particiones se generaron de manera estratificada, de tal modo que cada una de ellas contuviese la misma distribución de ejemplos de cada clase. Teniendo en cuenta un compromiso entre los valores de *g* y los tamaños de los conjuntos de entrenamiento empleados, se han marcado en negrita los mejores resultados, y en cursiva los segundos mejores.

Se puede comprobar como la estrategia de selección reduce considerablemente los conjuntos de datos, a la vez que mantiene o mejora los valores de *g*, sin que esto provoque una perdida de generalización por posible tendencia a la clase minoritaria y abandono de la clase mayoritaria (véanse los valores finales de acc^- y *acc* para cada

caso). Destáquense por ejemplo los casos de *breast-w*, donde con una reducción del conjunto de entrenamiento al 31.8% se siguen obteniendo mejoras en los valores de *acc* y *g*, o el de *vowel*, en el que con un 47.4% de los patrones iniciales se alcanza un valor de *g* del 99.7. Además, se puede observar como a mayores valores de M los conjuntos de datos están más equilibrados, y que la elección de este parámetro es decisiva para la obtención de submuestras óptimas, aunque con $M = 2$ y $M = 3$ se dieron buenos resultados en todos los problemas estudiados.

7. Conclusiones y trabajo futuro

En este trabajo se ha presentado un criterio que usa el voto mayoritario de los componentes de un PP para decidir qué patrones de entrenamiento son representativos a la hora de generar clasificadores independientes. Incorporándolo a una sencilla estrategia de selección se ha mostrado experimentalmente sobre PMCs (probablemente una de las herramientas más potentes en clasificación) una considerable reducción del tamaño de los conjuntos de entrenamiento, mientras se mantienen o incluso se mejoran sus capacidades de generalización.

Como posibles líneas de investigación futuras, podrían citarse el estudio de estrategias de selección más dependientes del clasificador final, así como la posibilidad de mantener cierta proporción de patrones redundantes durante la fase de filtrado, por el hecho de que estos ejemplos no estorban en el aprendizaje.

Referencias

[1] P. Auer, H. Burgsteiner, W. Maass. *Reducing Communication for Distributed Learning in Neural Networks*. Proceedings of ICANN'02, Lectures Notes in Computer Science 2415, pp 123-128, 2002.

[2] L. Breiman. *Bias, variance and arcing classifiers*. Tech. Report 460., Dept. of Statistics, University of California, Berkeley, 1996.

[3] N. Chawla, K. Bowyer, L. Hall, W. P. Kewelmeyer. *SMOTE: Synthetic Minority Over-sampling TEchnique*. In International Conference on Knowledge Based Computer Systems, 2000.

[4] J. R. Dorronsoro, F. Ginel, C. Sánchez, C. Santa Cruz. *Neural Fraud Detection in Credit Card Operations*. IEEE Transactions on Neural Networks, vol. 8, pp 827-834, 1997.

[5] W. Duch. *Support Vector Neural Training*. Submitted for publication, 2004.

[6] K. J. Ezawa, M. Singh, S. W. Norton. *Learning Goal Oriented Bayesian Networks for Telecommunications Management*. Proceedings of the International Conference on Machine Learning, ICML'96, pp 139-147, 1996.

[7] T. Fawcett, F. Provost. *Adaptative Fraud Detection*. Data Mining and Knowledge Discovery, no. 1, pp 291-316, 1996.

[8] T. Joachims. *Learning to Classify Text using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publisher, 2002.

[9] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, P. S. Meltzer. *Classification and Diagnostic of Cancers using Gene Expression Profiling and Artificial Neural Networks*. Nature Medicine, vol. 7, no. 6, pp 673-579, 2001.

[10] M. Kubat, S. Matwin. *Addressing the Curse of Imbalanced Training Sets: One-Sided Selection*. In Proceedings of the 14th International Conference on Machine Learning, ICML-97, pp 179-186, 1997.

[11] P. Murphy, D. Aha. *UCI Repository of Machine Learning Databases*. Tech. Report, University of California, Irvine, 1994.

[12] N. Nilsson. *The Mathematical Foundations of Learning Machines*. Morgan Kaufmann, 1990.

[13] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, C. Brunk. *Reducing Mis-classification Costs*. In Proceedings of the 11th International Conference on Machine Learning, pp 217-225, 1994.

[14] G. Weiss, F. Provost. *The Effect of Class Distribution on Classifier Learning: An Empirical Study*. Tech. Report ML-TR-44, Dept. of Computer Science, Rutgers University, 2001.

[15] D. L. Wilson. *Asymptotic Properties of Nearest Neighbour Rules Using Editing Data Sets*. IEEE Transactions on Systems, Man and Cybernetics, no. 2, pp 408-421, 1972.