

Selección dinámica de clasificadores para la estrategia Uno-contra-Uno: Evitando los clasificadores no competentes

Mikel Galar¹, Alberto Fernández², Edurne Barrenechea¹, Humberto Bustince¹,
and Francisco Herrera³

¹ Departamento de Automática y Computación,
Universidad Pública de Navarra, 31006, Pamplona, España
{mikel.galar, edurne.barrenechea, bustince}@unavarra.es

² Departamento de Ciencias de la Computación,
Universidad de Jaén, 23071, Jaén, España
alberto.fernandez@ujaen.es

³ Departamento de Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada, 18071, Granada, España
herrera@decsai.ugr.es

Resumen Una de las estrategias de descomposición más utilizada a la hora de afrontar problemas multi-clase es la estrategia Uno-contra-Uno, que divide el problema multi-clase original en tantos nuevos subproblemas como pares de clases puedan considerarse. Posteriormente, cada uno es afrontado por un clasificador binario independiente. Las nuevas instancias se clasifican combinando las salidas obtenidas por cada clasificador base. Sin embargo, esta forma de descomposición trae consigo ciertos problemas a la hora de la combinación, como es el denominado problema de los clasificadores no competentes. Se definen como aquellos cuya salida no es relevante para la clasificación de la instancia, es decir, no fueron aprendidos con instancias de la clase real del ejemplo a clasificar. En este trabajo proponemos un modelo de selección dinámica de clasificadores para la estrategia Uno-contra-Uno con la intención de evitar estos clasificadores no competentes, para lo que consideraremos el vecindario de la instancia a clasificar. Estudiamos la validez del modelo propuesto mediante un estudio experimental exhaustivo con diferentes clasificadores base y basados en un estudio estadístico adecuado.

Palabras clave: multi-clase, uno-contra-uno, descomposición, no competencia, selección de clasificadores

1. Introducción

Los problemas de clasificación se pueden dividir en dos tipos dependiendo del número de clases que consideran. Los problemas binarios tratan únicamente con un par de clases, mientras que los multi-clase consideran más de dos clases y por tanto son más generales. Normalmente, la clasificación con múltiples

clases es más difícil debido a la complejidad en el establecimiento de las fronteras de decisión. Adicionalmente a esta dificultad, inherente en los problemas multi-clase, algunos sistemas de clasificación están intrínsecamente diseñados para trabajar con solo dos clases, y su extensión a problemas con múltiples clases no está todavía establecida; por ejemplo, este es el caso de las Máquinas de Soporte Vectorial (*Support Vector Machines*, SVMs) [21] o el *Positive Definite Fuzzy Classifier* (PDFC) [4]. En estos casos, la forma natural de afrontar los problemas multi-clase es mediante técnicas de descomposición [17], que tratan de dividir el problema multi-clase original en problemas binarios más sencillos de resolver. Debemos destacar que este tipo de técnicas han mostrado su validez incluso en el caso en el que los clasificadores base utilizados puedan afrontar el problema multi-clase de forma directa [9].

En la literatura especializada existen multitud de estrategias de descomposición [17]. Entre ellas, la estrategia Uno-contra-Uno (*One-vs-One*, OVO) [15] es una de las más conocidas y utilizadas. En OVO se obtienen tantos nuevos subproblemas como pares de clases podamos obtener; cada uno de ellos es afrontado por un clasificador base independiente. Finalmente, una nueva instancia se clasifica sometiéndola a todos los clasificadores base y combinando sus salidas. Esta estrategia tan simple ha mostrado su potencial en trabajos previos [9], siendo además utilizada como estándar para afrontar problemas multi-clase con SVMs en los paquetes software más utilizados como WEKA [12], LIBSVM [3] o KEEL [2].

Una vez fijada la estrategia de descomposición, la forma en la que se agregan las salidas de los clasificadores es el siguiente factor clave para la clasificación. En [9] se llevó a cabo un extenso estudio experimental comparando las agregaciones del estado del arte para la estrategia OVO. Las estrategias analizadas iban desde las de estimación de probabilidades [24] hasta los modelos basados en relaciones de preferencia [7], entre otras. Uno de los problemas de OVO que más atención ha recibido por parte de los investigadores ha sido el de la región no clasificable cuando se considera la estrategia del voto simple. Sin embargo, esta atención no se ha visto reflejada en una mejora significativa de los resultados frente a otros modelos más simples como el voto ponderado [14]. Otro de los problemas mencionados en [9] es el de los clasificadores no competentes, cuyo estudio se indicó como una posible línea futura de trabajo prometedora ya que todavía no ha sido tratado de manera directa. La no competencia es inherente al método de descomposición de OVO; cada clasificador es entrenado únicamente con instancias de las dos clases que debe distinguir, ignorando las instancias del resto de clases. Es decir, son clases desconocidas para el clasificador y por tanto las salidas de estos clasificadores son irrelevantes para la clasificación de instancias de dichas clases. Sin embargo, estas salidas son agregadas de la misma forma que lo son las relevantes por lo que pueden llevar a etiquetar incorrectamente la instancia en determinadas situaciones.

Evidentemente, la competencia de los clasificadores no puede ser conocida a priori, ya que en ese caso el problema de clasificación estaría resuelto. Por ello, en este trabajo nuestra intención es presentar una nueva estrategia de agregación

basada en el concepto de Selección Dinámica de Clasificadores (SDC) [11,16] que pueda reducir el número de clasificadores no competentes que se utilizan en la combinación. De esta forma, podrían evitarse algunas clasificaciones erróneas debido a estos clasificadores. Mediante este modelo trataremos de considerar solo aquellos clasificadores que son probablemente competentes. Para ello, analizaremos el vecindario de la instancia a clasificar y a partir de este seleccionaremos el subconjunto de clasificadores para la fase de agregación.

En la literatura los métodos de SDC están principalmente dirigidos a conjuntos de clasificadores en los que todos los clasificadores son capaces de discernir entre todas las clases (cada uno está especializado en un área del espacio de entrada) [23]; por contra, su aplicación en la estrategia OVO todavía no ha sido estudiada, probablemente debido a que su aplicación es más difícil y está más restringida, ya que OVO establece el área de competencia de cada clasificador a priori, y no depende del espacio de entrada sino del de salida.

Para evaluar la validez del modelo propuesto hemos desarrollado un estudio experimental que es el mismo que hemos utilizado en [9], de tal forma que podemos basarnos en las conclusiones obtenidas en dicho trabajo para reducir la comparativa. Consideramos diecinueve conjuntos de datos del repositorio de KEEL [2,1], contrastamos los resultados mediante los test estadísticos adecuados [6,10] y comparamos nuestro modelo utilizando clasificadores base provenientes de diferentes paradigmas.

El resto del trabajo está organizado de la siguiente forma: en la Sección 2 recordamos la estrategia OVO e introducimos nuestro modelo dinámico. En la Sección 3 mostramos el marco experimental utilizado para la comparativa que llevamos a cabo en la Sección 4. Finalmente, la Sección 5 concluye este trabajo.

2. Selección dinámica de clasificadores para la estrategia Uno-contra-Uno

En esta sección introducimos nuestro modelo para evitar los clasificadores no competentes en OVO. Para ello, comenzamos recordando el funcionamiento de dicha estrategia (Subsección 2.1), posteriormente introducimos el modelo dinámico (Subsección 2.2) y mostramos un ejemplo sencillo de su funcionamiento (Subsección 2.3).

2.1. Estrategia Uno-contra-Uno

En OVO se divide un problema de m clases en $m(m-1)/2$ subproblemas binarios (todos los posibles pares de clases) que son afrontados por clasificadores base independientes. Una nueva instancia se clasifica obteniendo la salida de cada uno de los clasificadores. Cada clasificador que distingue un par de clases $\{C_i, C_j\}$ devuelve un grado de confianza $r_{ij} \in [0, 1]$ en favor de la clase C_i ($r_{ji} = 1 - r_{ij}$). Estas salidas pueden almacenarse en lo que se denomina matriz

de votos de la siguiente forma:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \quad (1)$$

Para establecer la clase a la que pertenece la instancia puede usarse cualquiera de las agregaciones existentes en la literatura [9] sobre la matriz de votos. Una de las estrategias más utilizadas por su robustez es la estrategia del voto ponderado (*Weighted Voting*, WV) [14] en donde cada confianza devuelta por los clasificadores base vota por la clase correspondiente y aquella que alcanza la mayor confianza total es predicha: $Class = \arg \max_{i=1, \dots, m} \sum_{1 \leq j \neq i \leq m} r_{ij}$

2.2. OVO Dinámico

Tal y como hemos expuesto anteriormente, nuestro objetivo es tratar de evitar aquellos clasificadores no competentes que pueden perjudicar a la decisión final del sistema cuando alguno de los clasificadores que distinguen la clase correcta han fallado (e incluso cuando no fallan, ver ejemplo en la Subsección 2.3). Para ello, consideramos el uso de modelos SDC; sin embargo, estos modelos de selección dinámica no se pueden aplicar directamente sobre OVO a diferencia de lo que ocurre con los conjuntos de clasificadores clásicos, ya que no podemos establecer las regiones en el espacio de entrada para cada clasificador base, ni podemos estimar sus precisiones locales. Por tanto, debemos adaptar estas técnicas al marco de trabajo de OVO, ya que los clasificadores base con los que trabajamos son únicamente competentes para las dos clases utilizadas en su entrenamiento. Evidentemente, no podemos restringir nuestra búsqueda de los clasificadores competentes a aquellos que han considerado la clase real en el entrenamiento, la cual lógicamente es desconocida; pero podemos intentar buscar un pequeño subconjunto de clases a las cuales la pertenencia de la instancia sea más probable. De este modo, podemos considerar una nueva matriz de votos donde consideramos únicamente los clasificadores entrenados con dichas clases; por tanto, podemos reducir el número de clasificadores en la clasificación eliminando aquellos de los que estamos suficientemente seguros de que no contribuyen a la decisión correcta, o que incluso pueden perjudicarla.

Siguiendo esta filosofía y de manera similar a modelos SDC previos [23], consideramos el uso de los vecinos de la instancia a ser clasificada para decidir si un clasificador es competente o no. Sin embargo, en nuestro caso, en vez de utilizar los vecinos para estimar la precisión local de los clasificadores, los utilizaremos para seleccionar las clases que tomarán parte en la matriz de votos reducida. El proceso de selección dinámica propuesto lo detallamos en los siguientes pasos:

1. Calcular los k vecinos más cercanos (k *Nearest Neighbors*, k NN) de la instancia a clasificar (k es un parámetro del método, actualmente fijado a $3 \cdot m$).

2. Obtener las clases presentes en dicho vecindario (si entre los k vecinos solo hay una clase, se aumenta el número de vecinos hasta un máximo de $2 \cdot k$, en cuyo caso utilizamos el OVO estándar).
3. Considerar una nueva matriz de votos en las que únicamente están presentes los clasificadores entrenados con pares de clases del subconjunto seleccionado.
4. Aplicar cualquiera de las agregaciones existentes para OVO [9].

La simplicidad del método es una importante ventaja, ya que consiste en utilizar k NN dinámicamente para hacer una pre-selección de clases sobre un amplio número de vecinos (de ahí el por qué de los valores de los parámetros). Gracias a ello, es difícil fallar una instancia por la eliminación de la clase correcta de la matriz de votos. Una vez llevada a cabo la pre-selección dinámica, podemos seguir utilizando cualquiera de las agregaciones existentes para OVO [9]. En nuestro caso, proponemos el uso del WV ya que su robustez ha sido probada tanto teórica [14] como empíricamente [9].

2.3. Ejemplo ilustrativo

Para mostrar como funciona el modelo propuesto, a continuación presentamos un ejemplo ilustrativo donde resolvemos el problema de la no competencia. Supongamos que tenemos que clasificar una instancia x , cuya clase real es c_1 y cuya matriz de votos OVO obtenida presentando dicha instancia a los clasificadores binarios es la siguiente:

$$R(x) = \begin{pmatrix} & \mathbf{c_1} & \mathbf{c_2} & \mathbf{c_3} & \mathbf{c_4} & \mathbf{c_5} \\ \mathbf{c_1} & - & 0,55 & 0,6 & 0,75 & 0,7 \\ \mathbf{c_2} & 0,45 & - & 0,4 & 1 & 0,8 \\ \mathbf{c_3} & 0,4 & 0,6 & - & 0,5 & 0,4 \\ \mathbf{c_4} & 0,25 & 0,0 & 0,5 & - & 0,1 \\ \mathbf{c_5} & 0,3 & 0,2 & 0,6 & 0,9 & - \end{pmatrix} \quad (2)$$

Si aplicamos la estrategia WV para obtener la clase final sobre $R(x)$, obtendríamos la clase c_2 como salida (Ecuación 3). Sin embargo, ninguno de los clasificadores que considera la c_1 en el entrenamiento ha fallado (primera fila); pero el resto de clasificadores que distinguen a la clase c_2 , a pesar de su no competencia (la cual es la fuente del fallo), han votado fuertemente por ella, y por eso en este caso sería la predicha.

$$R(x) = \begin{pmatrix} & \mathbf{c_1} & \mathbf{c_2} & \mathbf{c_3} & \mathbf{c_4} & \mathbf{c_5} & \mathbf{WV} \\ \mathbf{c_1} & - & 0,55 & 0,6 & 0,75 & 0,7 & 2,6 \\ \mathbf{c_2} & 0,45 & - & 0,4 & 1 & 0,8 & \mathbf{2,65} \\ \mathbf{c_3} & 0,4 & 0,6 & - & 0,5 & 0,4 & 1,9 \\ \mathbf{c_4} & 0,25 & 0,0 & 0,5 & - & 0,1 & 0,85 \\ \mathbf{c_5} & 0,3 & 0,2 & 0,6 & 0,9 & - & 2,1 \end{pmatrix} \quad (3)$$

En el caso de aplicar el modelo propuesto en vez del WV sobre la misma matriz de votos, debemos calcular primero los k vecinos más cercanos de x (donde $k = 3 \cdot 5$ clases = 15 vecinos). Supongamos que el subconjunto de clases en este vecindario es el formado por las clases $\{c_1, c_4, c_5\}$. Por consiguiente, eliminamos de la matriz de votos aquellos clasificadores que no consideran pares de clases del subconjunto (Ecuación 4), es decir, cualquier clasificador que haya sido entrenado

con instancias de las clases $\{c_2, c_3\}$. Finalmente, aplicando el WV sobre la nueva matriz de votos. En este caso, la clase predicha (c_1) es la correcta.

$$R_{dyn}(x) = \left(\begin{array}{ccccc|c} & c_1 & c_2 & c_3 & c_4 & c_5 & WV \\ c_1 & - & 0.55 & 0.6 & 0.75 & 0.7 & 1.45 \\ c_2 & 0.45 & - & 0.4 & 1 & 0.8 & - \\ c_3 & 0.4 & 0.6 & - & 0.5 & 0.4 & - \\ c_4 & 0.25 & 0.0 & 0.5 & - & 0.1 & 0.35 \\ c_5 & 0.3 & 0.2 & 0.6 & 0.9 & - & 1.2 \end{array} \right) \quad (4)$$

3. Marco Experimental

En esta sección presentamos el marco experimental utilizado para el desarrollo de la comparativa experimental de la Sección 4. Debemos destacar que todo el marco experimental utilizado coincide con el de [9] y por tanto, partimos de las conclusiones obtenidas en dicho estudio para llevar a cabo un estudio experimental directo, en el que comparamos nuestro modelo contra los mejores encontrados en dicho trabajo. Primero describimos los clasificadores considerados y sus parámetros en la Subsección 3.1. Después, en la Subsección 3.2 recordamos las mejores estrategias de agregación para cada clasificador base en [9]. Posteriormente, detallamos los conjuntos de datos y los tests estadísticos considerados Subsección 3.3.

3.1. Clasificadores base y parámetros

Hemos considerado diferentes clasificadores base de diferentes paradigmas para comparar nuestro modelo en situaciones diferentes. Los algoritmos de aprendizaje utilizados son los siguientes: Las máquinas de soporte vectorial **SVM** [21], el árbol de decisión **C4.5** [20], los k vecinos más cercanos **kNN** [18], el aprendizaje de reglas **Ripper** [5], y el clasificador difuso **PDFC** [4].

La mayoría de los métodos de combinación para OVO llevan a cabo su predicción en base a la confianza dada por los clasificadores base para la decisión. En este trabajo obtenemos las confianzas para cada uno de los anteriores de la siguiente forma:

- **SVM**: Modelo logístico para la estimación de probabilidades [19].
- **C4.5**: Pureza de la hoja que lleva a cabo la predicción.
- **kNN**: Confianza = $\frac{\sum_{l=1}^k \frac{e_l}{d_l}}{\sum_{l=1}^k \frac{1}{d_l}}$ donde d_l es la distancia entre la instancia y el vecino l -ésimo y $e_l = 1$ si el vecino l es de la clase y 0 en otro caso.
- **Ripper**: Pureza de la regla que realiza la predicción.
- **PDFC**: Confianza total para la clase predicha.

En caso de que ocurran empates, la clase mayoritaria es la predicha, si el empate continúa, la clase se selecciona de manera aleatoria.

Los parámetros utilizados en cada clasificador base pueden verse en la Tabla 1. Estos parámetros son comunes para todos los problemas y han sido seleccionados de acuerdo a las recomendaciones de los autores. Hacer notar que usamos

dos configuraciones de SVM para comprobar la robustez de la propuesta. Finalmente, para el modelo dinámico propuesto utilizamos la distancia Euclidea para la búsqueda de los vecinos más cercanos (excepto cuando el conjunto de datos contiene valores nominales que utilizamos la distancia HVDM).

Tabla 1. Especificación de parámetros.

Algoritmo	Parámetros
SVM _{Poly}	C = 1.0, Tolerancia = 0.001, Epsilon = 1.0E-12, Kernel Type = Polinomial, Grado polinomio = 1
SVM _{Puk}	C = 100.0, Tolerancia = 0.001, Epsilon = 1.0E-12 Kernel Type = Puk, PukKernel ω = 1.0, PukKernel σ = 1.0
C4.5	Poda = True, Nivel confianza = 0.25, Mínimo número de item-sets por hoja = 2
3NN	$k = 3$, Distancia = Heterogeneous Value Difference Metric (HVDM)
Ripper	Tamaño del subconjunto = 66 %, Repeticiones de la optimización = 2
PDFC	C = 100.0, Tolerancia = 0.001, Epsilon = 1.0E-12, Kernel Type = Polinomial Grado Polinomio = 1, Tipo PDRF = Gaussiano

3.2. Agregaciones consideradas

Utilizamos como agregaciones representativas para cada clasificador las mismas seleccionadas en [9]. La única excepción es con las SVMs, ya que no se encontraron diferencias significativas entre los métodos y el modelo más utilizado es el de Wu *et al.* [24], que además nos permite utilizar las mismas matrices OVO para todos los métodos. Por tanto, las agregaciones para cada clasificador son las siguientes: **SVM** – PE (Wu *et al. Probability Estimates by Pairwise Coupling* [24]). **C4.5** – WV (*Weighted Voting strategy*). **kNN** – ND (*Non-Dominance criterion* [7]). **Ripper** – WV (*Weighted Voting strategy*). **PDFC** – PC (*Probability Estimates by Pairwise Coupling* [13]).

Una descripción completa y detallada de los métodos puede consultarse en [8]. Debemos recalcar que todos los métodos utilizan exactamente las mismas matrices de votos y que por tanto todas las diferencias mostradas en el estudio experimental se deben únicamente a la estrategia de agregación.

3.3. Conjuntos de datos y evaluación

Consideramos los mismos diecinueve conjuntos de datos del repositorio KEEL [1] utilizados en [9], cuyas propiedades se muestran en la Tabla 2. Utilizamos un modelo de validación cruzada de 5 particiones para obtener los resultados de precisión de cada método. Para llevar a cabo una evaluación honesta de los resultados, utilizamos test estadísticos no paramétricos [6,10]. Dado que realizamos comparativas por pares utilizaremos el test de Wilcoxon [22].

4. Estudio experimental

En esta sección vamos a estudiar la utilidad del modelo propuesto. Para ello, comparamos nuestro modelo dinámico frente a las mejores estrategias del estado del arte para cada clasificador base. Nuestro objetivo es investigar si el hecho de evitar los clasificadores no competentes se traduce en una mejora de los

Tabla 2. Descripción de los conjuntos de datos.

C. datos	#Ej.	#Atr.	#Num.	#Nom.	#Cl.
Car	1728	6	0	6	4
Lymphography	148	18	3	15	4
Vehicle	846	18	18	0	4
Cleveland	297	13	13	0	5
Nursery	1296	8	0	8	5
Page-blocks	548	10	10	0	5
Autos	159	25	15	10	6
Dermatology	358	34	1	33	6
Flare	1066	11	0	11	6
Glass	214	9	9	0	7
Satimage	643	36	36	0	7
Segment	2310	19	19	0	7
Shuttle	2175	9	9	0	7
Zoo	101	16	0	16	7
Ecoli	336	7	7	0	8
Led7digit	500	7	0	7	10
Penbased	1100	16	16	0	10
Yeast	1484	8	8	0	10
Vowel	990	13	13	0	11

resultados; o por contra, si al tratar de evitarlos mediante el mecanismo dinámico estamos también eliminando las clases que deben ser predichas del proceso de decisión.

La Tabla 3 muestra los resultados de precisión en test de los diferentes métodos. Mostramos para cada clasificador base el resultado de la mejor agregación anterior y el de la estrategia dinámica (Dyn) con el WV. Podemos comprobar como la propuesta funciona aparentemente bien en todos los clasificadores base, aumentando la precisión media en todos ellos y destacando en un gran número de conjuntos de datos. Sin embargo, para obtener conclusiones relevantes debemos llevar a cabo el estudio estadístico adecuado, en este caso, mediante el test de Wilcoxon, cuyos resultados se muestran en la Tabla 4.

Tabla 3. Resultados de precisión media en test para las agregaciones representativas y la estrategia dinámica (con WV) para cada clasificador base.

C. datos	3NN		C4.5		Ripper		SVM _{Poly}		SVM _{Puk}		PDFC	
	ND	Dyn ^{WV}	WV	Dyn ^{WV}	WV	Dyn ^{WV}	PE	Dyn ^{WV}	PE	Dyn ^{WV}	PC	Dyn ^{WV}
autos	76.75	77.38	81.17	79.92	78.65	79.27	74.80	75.42	68.53	67.88	76.71	78.61
car	92.82	93.06	93.00	93.29	91.32	92.36	92.71	92.65	63.60	84.72	99.88	99.94
cleveland	56.58	56.58	51.53	52.53	48.45	49.81	58.25	57.59	45.09	45.42	52.83	53.85
derma.	89.96	93.58	96.37	98.31	93.02	93.30	94.13	94.69	96.09	96.09	84.36	92.18
ecoli	80.96	80.66	79.47	80.07	75.89	75.60	77.69	77.99	75.31	75.90	84.36	82.75
flare	72.14	71.95	74.20	74.29	72.61	72.61	74.67	75.33	69.42	72.89	72.98	72.98
glass	73.38	73.38	70.53	71.47	74.30	74.30	61.26	62.66	70.60	71.09	68.25	70.11
led7digit	72.60	72.80	72.20	72.60	70.00	71.60	73.00	74.00	70.20	71.40	71.80	72.60
lymph.	83.72	83.72	73.63	76.30	76.28	77.63	81.68	81.68	80.34	80.34	78.94	78.94
nursery	92.52	92.52	89.04	88.89	89.43	89.28	91.90	91.90	81.33	89.12	96.84	96.84
pageblocks	94.70	94.88	95.61	95.79	95.25	95.25	94.70	94.52	94.16	94.34	95.43	95.43
penbased	96.18	96.27	90.64	90.73	89.91	90.82	95.27	95.64	97.82	97.91	98.27	98.27
satimage	86.47	86.32	81.65	82.74	79.78	80.40	84.14	83.67	84.92	85.08	87.41	87.10
segment	96.71	96.97	97.06	96.97	95.84	96.41	92.55	92.60	97.10	97.06	96.62	96.54
shuttle	99.54	99.63	99.72	99.68	99.49	99.68	96.37	97.52	99.72	99.68	97.47	98.21
vehicle	71.40	71.40	71.39	72.81	71.04	71.75	72.46	73.29	80.49	80.61	83.57	83.69
vowel	95.86	95.86	80.00	81.21	78.99	78.48	69.90	70.30	99.39	99.39	97.98	97.98
yeast	54.72	54.52	59.91	59.57	56.00	56.07	59.10	58.96	56.54	56.27	59.10	59.03
zoo	92.10	94.10	93.10	93.10	94.10	95.10	95.05	96.05	84.19	85.19	97.05	97.05
Mean	83.11	83.45	81.59	82.12	80.54	81.04	81.03	81.39	79.73	81.60	84.12	84.85

Los resultados de los test estadísticos son claros, siendo la estrategia dinámica significativamente mejor que las representativas para cada clasificador base en cinco de los seis casos estudiados (todos menos 3NN), con p-valores muy bajos. En el caso de 3NN, aunque el test no se rechaza, los rangos están a favor de la estrategia dinámica con un p-valor bajo. Por tanto, estos resultados muestran la superioridad de la estrategia dinámica frente a los métodos clásicos. Además

Tabla 4. Comparativa de agregaciones mediante el test de Wilcoxon. R^+ corresponde a la suma de rangos para la agregación clásica y R^- para la dinámica.

Clasificador	Comparativa	R^+	R^-	Hipótesis	p-valor
3NN	Dyn ^{WV} vs. ND	133.5	56.5	No Rechazada	0.140146
C4.5	Dyn ^{WV} vs. WV	153.5	36.5	Rechazada para Dyn ^{WV} at 5 %	0.017621
Ripper	Dyn ^{WV} vs. WV	162.5	27.5	Rechazada para Dyn ^{WV} at 5 %	0.005684
SVM _{Poly}	Dyn ^{WV} vs. PE	149.5	40.5	Rechazada para Dyn ^{WV} at 5 %	0.024520
SVM _{Puk}	Dyn ^{WV} vs. PE	154.0	36.0	Rechazada para Dyn ^{WV} at 5 %	0.015086
PDFC	Dyn ^{WV} vs. PC	138.0	52.0	Rechazada para Dyn ^{WV} at 5 %	0.040860

queda claro que el evitar algunos de los clasificadores no competentes puede llevar a mejoras significativas de los resultados, lo que a su vez pone de manifiesto este problema en la estrategia OVO.

5. Conclusiones

En este trabajo hemos presentado una nueva estrategia para agregar los resultados de los clasificadores base en la estrategia OVO mediante un modelo SDC. De esta forma, logramos evitar que las decisiones de alguno de los clasificadores no competentes perjudique la clasificación. Este modelo utiliza el vecindario de la instancia a clasificar para establecer la competencia de cada clasificador, pasando a utilizar únicamente aquellos que, con nuestro método, son competentes.

La nueva estrategia ha mostrado su utilidad a pesar de su simplicidad en el estudio experimental llevado a cabo. A su vez, este hecho ha puesto de manifiesto la importancia del problema de los clasificadores no competentes en OVO, cuyo tratamiento puede llevar a mejoras significativas en los resultados.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Educación y Ciencia bajo los proyectos TIN2010-15055 y TIN2011-28488 y el plan Andaluz de investigación P10-TIC-6858.

Referencias

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Mult.-Valued Logic Soft Comput.* 17, 255 – 287 (2011)
2. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J., Herrera, F.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Comput.* 13(3), 307–318 (2009)
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 27:1–27:27 (2011)
4. Chen, Y., Wang, J.Z.: Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions Fuzzy Systems* 11(6), 716–728 (2003)

5. Cohen, W.W.: Fast effective rule induction. In: ICML'95: Proceedings of the Twelfth International Conference on Machine Learning. pp. 1–10 (1995)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)
7. Fernández, A., Calderón, M., Barrenechea, E., Bustince, H., Herrera, F.: Solving mult-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations. *Fuzzy Sets Syst.* 161(23), 3064 – 3080 (2010)
8. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: Aggregation schemes for binarization techniques. methods' description. Tech. rep. (2011), <http://sci2s.ugr.es/ovo-ova/AggregationMethodsDescription.pdf>
9. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recogn.* 44(8), 1761 – 1776 (2011)
10. García, S., Herrera, F.: An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *J. Mach. Learn. Res.* 9, 2677–2694 (2008)
11. Gunes, V., Ménard, M., Loonis, P., Petit-Renaud, S.: Combination, cooperation and selection of classifiers: A state of the art. *Int. J. Pattern Recogn.* 17(8), 1303 – 1324 (2003)
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *SIGKDD Explor. Newsl.* 11, 10 – 18 (2009)
13. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. *Ann. Statist.* 26(2), 451–471 (1998)
14. Hüllermeier, E., Vanderlooy, S.: Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. *Pattern Recogn.* 43(1), 128–142 (2010)
15. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: A stepwise procedure for building and training a neural network. In: Fogelman Soulié, F., Héroult, J. (eds.) *Neurocomputing: Algorithms, Architectures and Applications*, NATO ASI Series, vol. F68, pp. 41–50. Springer-Verlag (1990)
16. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience (2004)
17. Lorena, A.C., Carvalho, A.C., Gama, J.M.: A review on the combination of binary classifiers in multiclass problems. *Artif. Intell. Rev.* 30(1-4), 19–37 (2008)
18. McLachlan, G.J.: *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley and Sons (2004)
19. Platt, J.C.: *Fast training of support vector machines using sequential minimal optimization*. MIT Press, Cambridge, MA, USA (1999)
20. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. San Mateo-California: Morgan Kaufmann Publishers, 1st edn. (1993)
21. Vapnik, V.: *Statistical Learning Theory*. New York: Wiley (1998)
22. Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics Bull.* 1(6), 80–83 (1945)
23. Woods, K., Philip Kegelmeyer, W., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. Pattern Anal. Mach. Intell.* 19(4), 405 – 410 (1997)
24. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.* 5, 975–1005 (2004)