Capítulo 6

Máquinas de Aprendizaje *K*-SVCR en Multiclasificación

"Las ciencias aplicadas no existen, sólo las aplicaciones de la ciencia"
Louis Pasteur.

Para comenzar

La finalidad última de la máquina presentada en el Capítulo 5, aún siendo definida como una entidad independiente, es su utilización en problemas de clasificación multiclase siguiendo una metodología de descomposición y reconstrucción. Su implementación en este tipo de problemas es sencilla y similar a la realizada en las formulaciones estándares. Sin embargo, la utilización de todas las clases en el entrenamiento de todos los nodos de dicotomía, aún aumentando el tiempo de cómputo, provoca la aparición de una cierta redundancia que le confiere a esta nueva formulación una tolerancia a fallos muy superior a la obtenida con arquitecturas convencionales. Para hacer plausible esta robustez se hace necesario definir un elemento intérprete y un elemento de combinación de predicciones que favorezcan esta característica.

Hasta el momento, tan sólo el uso de la teoría de 'códigos de corrección de error' (ECOC) había sido estudiado para aumentar la tolerancia de las máquinas de multiclasificación, por lo que es necesario realizar una comparativa con respecto a esta tipología de máquinas. Si bien el estudio respecto a las técnicas estándares se realiza de forma completamente analítica por lo que se puede llegar a conclusiones del todo formales, la introducción de la aleatoriedad en la elección de los códigos de error en la teoría ECOC obliga a que este estudio deba ser realizado, en algunos casos, de forma empírica, por lo que los datos obtenidos pueden poseer una cierta desviación, pero sí puede asegurarse analíticamente que el comportamiento robusto de la teoría ECOC

sobre máquinas triclase es cualitativamente superior a aquél obtenido sobre máquinas biclase.

6.1 Esquema de Descomposición para Multiclasificación

A pesar de la entidad propia que posee la nueva máquina de clasificación como una extensión al caso triclase de una SVMC estándar, su definición ha estado motivada por la voluntad de obtener una nueva arquitectura de multiclasificación {descomposición, reconstrucción} que mejore en la medida de lo posible las ya existentes. Se hace necesaria así la definición del esquema de descomposición basado en la nueva máquina y una comparativa con los descritos anteriormente. Utilizando la notación especificada en el Capítulo 4, con la salvedad de generalizar la asignación de patrones salida de entrenamiento para cada dicotomía al caso ternario, $t_p^k \in \{-1, 0, 1\}$, se realiza la siguiente definición.

Definición 6.1. *Un k-ésimo nodo de dicotomía, dentro de una arquitectura de máquina de clasificación multiclase se denomina* 1,1-v-r SVMC *— del inglés, one-versus-one-versus rest — si es una K-SVCR entrenada con etiquetas positivas $t_p^k = 1$ para los ℓ_i patrones de entrenamiento de la i-ésima clase etiquetada θ_i, $\mathcal{T}_+^k = \mathcal{T}_i$, con etiquetas negativas $t_p^k = -1$ para los ℓ_j patrones de entrenamiento de la j-ésima clase etiquetada θ_j, $\mathcal{T}_-^k = \mathcal{T}_j$ y con etiqueta nula $t_p^k = 0$ para los $\ell - (\ell_i + \ell_j)$ patrones, $\mathcal{T}_0^k = \mathcal{T}^* \backslash (\mathcal{T}_i \cup \mathcal{T}_j)$, de las $K - 2$ clases restantes.*

El método de descomposición asociado a este tipo de dicotomías consiste en situar $L = K(K-1)/2$ clasificadores ternarios del estilo 1,1-v-r en paralelo, cada nodo siendo entrenado sobre todo el conjunto de entrenamiento modificado, $\mathcal{T}^k = \mathcal{T}^*$.

El clasificador ternario K-SVCR mejora las estructuras estándares empleadas en el método de descomposición de una arquitectura de clasificación multiclase: la máquina de aprendizaje realiza esencialmente una partición entre dos clases, tal como lo hace el esquema 1-v-1, siendo capaz ahora de dar siempre una respuesta con sentido al obligar a etiquetar con 0 las otras clases implicadas en la clasificación. Más aún, cada 1,1-v-r SVMC es entrenada sobre todo el conjunto de patrones de aprendizaje, como en el esquema 1-v-r, de forma que las salidas de todos los clasificadores pueden ser considerados en la fase de reconstrucción. En la Tabla 6.1 se muestra una comparativa de la complejidad del nuevo método de descomposición respecto a aquellos que son estándar, de nuevo ilustrada para el caso de $\ell = 1000$ patrones de entrenamiento, divididos en $K = 10$ clases, teniendo cada clase la misma cantidad de representantes, $\ell_i = \ell/K = 100, \forall i$. Observando los resultados de la Tabla adjunta, puede afirmarse que la nueva máquina es más compleja que las anteriores en un factor menor a 2. Aunque no se ha obtenido la máquina que sería ideal, el coste computacional no es

	Nodos de Dicotomía	Variables	Restricciones $\leq$
1-v-r	$K = 10$	$\ell = 1000$	$2(V+1) = 2002$
1-v-1	$\binom{K}{2} = 45$	$2\ell/K = 200$	$2(V+1) = 402$
1,1-v-r SVMC	$\binom{K}{2} = 45$	$2(K-1)\ell/K = 1800$	$2(V+1) = 3600$
Deseable	$\leq \binom{K}{2} = 45$	$\ell = 1000$	$2(V+1) = 2002$

Tabla 6.1: Comparativa sobre la complejidad del nuevo método de descomposición respecto a las descomposiciones estándares.

	Nodos de Dicotomía	Variables	Restricciones $\leq$
ECOC est.	$\leq 2^{(K-1)} - 1 = 511$	$\ell = 1000$	$2(V+1) = 2002$
ECOC gener.	$\leq K\left(2^{(K-1)} - 1\right) = 5110$	$\leq \ell = 1000$	$2(V+1) \leq 2002$
ECOC capas	$K(K-1) = 90$	$2\ell/K, \ell$	$2(V+1)$
1,1-v-r SVMC	$\binom{K}{2} = 45$	$2(K-1)\ell/K$	$2(V+1) = 3600$
Deseable	$\leq \binom{K}{2} = 45$	$\ell = 1000$	$2(V+1) = 2002$

Tabla 6.2: Comparativa sobre la complejidad del nuevo método de descomposición respecto a las descomposiciones basadas en la teoría ECOC.

significativo, menos aún si es comparado con los clasificadores "todas las clases a la vez". Por otra parte, éste será el único precio a pagar para conseguir una estructura mucho más robusta. No será necesario construir arquitecturas de multiclasificación con descomposiciones redundantes para obtener un clasificador global más tolerante a fallos, como sí era necesario en las estructuras con metodología ECOC. Incluso yendo algo más allá, cada 1,1-v-r SVMC podría también ser interpretada como una descomposición 2-v-r, si se consideran las clases i-ésima y j-ésima implicadas inicialmente de forma individual, como un global frente a todas las otras clases.

Utilizando los valores ejemplo de la Tabla 6.1, se obtiene la comparativa mostrada en la Tabla 6.2. Si bien los problemas de optimización asociados continúan siendo mayores que los estándares, la nueva arquitectura mejora a aquella que usa la teoría ECOC para obtener mejoras de robustez, pues el número de dicotomías está fijado en $\binom{K}{2}$.

Con respecto a las máquinas que consideran "todas las clases a la vez", es cierto que es necesario solucionar bastantes más que un único QP problema con la nueva metodología, pero el tamaño de los problemas es muy inferior.

En [Platt, 1999a] se observa de forma empírica que el entrenamiento de una SVMC crece superlinealmente con la talla del conjunto de entrenamiento, ℓ, de acuerdo a

una ley exponencial

$$T_{SVM} = c \cdot \ell^{\gamma} \quad : \quad \gamma \approx 2\,, \tag{6.1}$$

lo que permite obtener para un caso típico donde $\gamma = 2$, los siguientes tiempos de entrenamiento:

$$\begin{array}{l} T_{1-v-r} = c \cdot \ell^2 \cdot K \\ T_{1-v-1} = c \cdot \ell^2 \cdot 2 \end{array} \quad . \tag{6.2}$$

Si se aproxima la talla del conjunto de entrenamiento por el número de variables implicadas en el problema de optimización, se obtiene la siguiente aproximación de tiempos de cálculo

$$\begin{array}{l} \widetilde{T}_{1-v-r} = c \cdot \ell^2 \cdot K \\ \widetilde{T}_{1-v-1} = c \cdot \ell^2 \cdot 2\dfrac{(K-1)}{K} \end{array} \quad , \tag{6.3}$$

que permitiría extender el estudio empírico sobre tiempo de entrenamiento al resto de esquemas de descomposición tratados

$$\begin{array}{r} \widetilde{T}_{1,1-v-r} = c \cdot \ell^2 \cdot \dfrac{2\,(K-1)^3}{K^2} \\ \widetilde{T}_{todos} = c \cdot \ell^2 \cdot (K-1)^2 \end{array} \quad . \tag{6.4}$$

Se observa que en todas las expresiones aparece el término $c \cdot \ell^2$, por lo que el tiempo de entrenamiento final dependería únicamente del número de clases en un factor t que para el ejemplo que se está siguiendo en esta exposición se podría calcular como

$$\begin{array}{ll} t_{1-v-r} = 10, & t_{1-v-1} = 1.8 \\ t_{1,1-v-r} = 14.58, & t_{todos} = 81 \end{array} \quad . \tag{6.5}$$

Estos factores, al ser una aproximación empírica, se encuentran algo distorsionados, pero en las pocas comparativas que se han hallado en la literatura sobre tiempos de entrenamiento para métodos de descomposición, parecen confirmarse magnitudes similares pues en [Bredensteiner and Bennett, 1999] se obtiene sobre el 'benchmark' *US Postal Service* $t_{todos} = 2.96 \cdot t_{1-v-r}$ si el problema es QP, como las SVM, y $t_{todos} = 23.23 \cdot t_{1-v-r}$ si el problema asociado es lineal, LP; y en [Platt et al., 2000] $t_{1-v-r} = 12.17 \cdot t_{1-v-1}$ sobre el mismo *USPS*, y $t_{1-v-r} = 5.07 \cdot t_{1-v-1}$ sobre el 'benchmark' *UCI Letter*. La conclusión que puede extraerse de modo aproximado es que el nuevo método de descomposición 1,1-v-r SVMC es de una complejidad mayor pero de orden similar al método 1-v-r SVMC aunque construya más conjuntos de entrenamiento, mientras que es de una complejidad mucho menor que la descomposición "todas las clases a la vez" aunque sea necesaria sólo una máquina.

Aunque los métodos basados en teoría ECOC no pueden ser estudiados de forma exacta debido a su aleatoriedad en la elección de nodos de dicotomía, puede asegurarse atendiendo a su modo de selección de las clases que intervienen en estos nodos que

su complejidad es superior a aquella del método 1-v-r SVMC, por lo que, en el mejor de los casos, puede pensarse que poseen tiempos de entrenamiento similares a 1,1-v-r SVMC.

Sin duda el método que ha salido mejor parado de esta comparativa, con diferencia, ha resultado ser el 1-v-1 SVMC, por lo que el método en árbol DAGSVM que posee las ventajas adicionales de estar basado en un resultado teórico de generalización y de necesitar sólo $K-1$ evaluaciones sobre nodos del árbol para obtener la respuesta final parece el método hasta el momento más indicado de trabajo. Sin embargo, de las seis propiedades deseables para una máquina de multiclasificación enunciadas al principio del Capítulo 5, las arquitecturas basadas en máquinas 1-v-1 incumplen dos importantes, a saber, las dicotomías entrenadas tendrán un poder de generalización con mucha varianza debido a que han sido entrenadas sobre un conjunto reducido de patrones, y su tolerancia a fallos en las respuestas parciales es nula. La nueva máquina, en cambio, es algo más compleja que las estándares, aunque de coste computacional de un orden similar, pero permite que cada dicotomía trabaje sobre todo el conjunto de entrenamiento y será probada su robustez ante fallos, lo que la convierte en una opción muy válida para multiclasificar.

6.2 Método de Reconstrucción

La máquina K-SVCR se ha mostrado como un algoritmo apropiado para tratar problemas de separación de dos clases en un entorno multiclase sin dejar de tener en cuenta la disposición espacial de los patrones del resto de clases, por lo que son particularmente útiles en esquemas de descomposición en paralelo. La segunda gran ventaja que aportan este tipo de máquinas es la información que se encierra en su respuesta. Sin necesidad de recurrir a su valor numérico, sólo a su signo, las respuestas de tipo $\{-1, 0, +1\}$ permiten múltiples interpretaciones sobre la información emitida.

6.2.1 Interpretación de las Predicciones

Recordando la notación del Capítulo 4, sea $z^k = f_k(\mathbf{x})$ la respuesta numérica facilitada por el hiperplano separador construido por el nodo de dicotomía (i,j)-v-r SVMC, f_k, a una entrada $\mathbf{x} \in \theta_m$, y sea $s^k = h(f_k(\mathbf{x})) = sign\left(z^k\right) \in \{-1, 0, +1\}$ el signo que retorna la función de decisión. ¿Cómo debe traducir el elemento intérprete, $\Theta\left(s^k\right)$, la respuesta signatoria?¿Cómo deben ser combinadas posteriormente estas traducciones por el elemento combinatorio, Ψ?

Definición 6.2. *Dado un nodo de dicotomía (i,j)-v-r SVMC, f_k, se define como*

- interpretación positiva precisa*, aquella que tiene por elemento intérprete*

$$\Theta_{PP}\left(s^k\right) = \begin{cases} \theta_i & si\ s^k = +1 \\ \theta_j & si\ s^k = -1 \\ Y \backslash \{\theta_i, \theta_j\} & si\ s^k = 0 \end{cases}, \tag{6.6}$$

- interpretación negativa precisa*, aquella que tiene por elemento intérprete*

$$\Theta_{NP}\left(s^k\right) = \neg\mathcal{C}\left(\Theta_{PP}\left(s^k\right)\right) = \begin{cases} \neg\left(Y \backslash \theta_i\right) & si\ s^k = +1 \\ \neg\left(Y \backslash \theta_j\right) & si\ s^k = -1 \\ \neg\{\theta_i, \theta_j\} & si\ s^k = 0 \end{cases}, \tag{6.7}$$

  donde $\mathcal{C}$ indica el conjunto complementario y $\neg$ es el operador negación,

- interpretación mixta precisa*, aquella que tiene por elemento intérprete*

$$\Theta_{MP}\left(s^k\right) = \begin{cases} \theta_i & si\ s^k = +1 \\ \theta_j & si\ s^k = -1 \\ \neg\{\theta_i, \theta_j\} & si\ s^k = 0 \end{cases}, \tag{6.8}$$

- interpretación positiva imprecisa*, aquella que tiene por elemento intérprete*

$$\Theta_{PI}\left(s^k\right) = \begin{cases} \theta_i & si\ s^k = +1 \\ \theta_j & si\ s^k = -1 \\ Y & si\ s^k = 0 \end{cases}, \tag{6.9}$$

- interpretación negativa imprecisa*, aquella que tiene por elemento intérprete*

$$\Theta_{NI}\left(s^k\right) = \neg\mathcal{C}\left(\Theta_{NP}\left(s^k\right)\right) = \begin{cases} \neg\left(Y \backslash \theta_i\right) & si\ s^k = +1 \\ \neg\left(Y \backslash \theta_j\right) & si\ s^k = -1 \\ \emptyset & si\ s^k = 0 \end{cases}. \tag{6.10}$$

¿Qué representa cada una de estas interpretaciones? En el primer caso, $\Theta_{PP}\left(s^k\right)$, la interpretación asegura la pertenencia del elemento a una cierta clase si la salida signatoria es no nula, mientras que en caso nulo afirma que pertenece a alguna de las restantes. El intérprete está seguro de que todos los hiperplanos separadores son correctos de forma precisa. En el caso de interpretación negativa precisa, el intérprete posee la misma seguridad en los hiperplanos pero lo interpreta enunciándolo en forma negativa.

En la tercera interpretación, la mixta precisa, ante una respuesta no nula el intérprete asegura la pertenencia de la entrada a una clase, pero ante una respuesta nula tan sólo se atreve a afirmar que la respuesta no es ninguna de las clases con

separación única, θ_i y θ_j. Este tipo de interpretación, aún siendo precisa, recoge la posibilidad de existencia de otras clases de patrones que no han sido catalogadas en el conjunto de salidas inicial $\mathcal{Y}$, y entiende que el clasificador las recogerá dentro del grupo de "otras clases" que representa la etiqueta 0. Será éste el intérprete elegido para realizar la combinación de respuestas.

Por último, la cuarta y quinta definición recogen una forma de interpretación imprecisa en el sentido que suponen unos hiperplanos de separación de las clases únicas muy seguros de sus respuestas, pero posiblemente existan elementos de esos grupos que hayan sido etiquetados 0. Estas situaciones resultan adecuadas cuando alguna de las clases está simbolizando un peligro o coste importante. Supóngase que se entran los resultados de unos análisis clínicos a una máquina de aprendizaje para asesorar al médico sobre la posible enfermedad del paciente. Si alguna de las enfermedades contenidas en el conjunto de clases representa un grave trastorno para el paciente, el doctor deseará estar seguro de que la respuesta en ese sentido de la máquina sea lo más fiable posible y preferirá como respuesta máquina una ambigüedad antes que un diagnóstico equivocado. Se verá más adelante como el aumento en la seguridad de la respuesta cuando el resultado ha de ser una cierta clase en particular puede ser conseguido con la interpretación mixta precisa, Θ_{MP}, si el parámetro δ de la máquina K-SVCR es elegido apropiadamente grande.

Por tanto, la interpretación mixta precisa sobre máquinas K-SVCR recoge tres características que engloban al resto de interpretaciones:

1. Asume precisión en las respuestas de los nodos de dicotomía.
2. Permite trabajar con conjuntos de clases que no son completos.
3. Permite controlar la precisión de trabajo sobre las clases que se deseen mediante el parámetro δ del nodo de dicotomía.

Se comprobará a continuación que no sólo posee estas tres propiedades sino que además facilita la creación de un elemento de combinación que convierte el método de reconstrucción en muy robusto.

6.2.2 Elemento de Combinación y Análisis de Robustez

Puesto que se cuenta de forma directa con votos positivos y negativos, la combinación más natural de las interpretaciones de cada nodo de dicotomía es realizar un recuento para cada clase con la suma de votos emitidos y asignar como clase aquella que obtenga un número mayor de votos, sin necesidad de crearlos de modo artificial como se realizaba en la clasificación "por parejas".

Definición 6.3. *Sea* $\left\{\Theta\left(s^k\right)\right\}_{k=1}^{L}$ *el conjunto de interpretaciones de los* $L = \binom{K}{2}$ *nodos de dicotomía 1,1-v-r SVMC, de la forma (6.8). Si* Ψ_r *es el cómputo de votos positivos y negativos obtenidos por la clase* θ_r *a partir del elemento intérprete mixto preciso, entonces se define como* elemento de combinación de la arquitectura 1,1-v-r SVMC *la función*

$$\Psi\left(\left\{\Theta\left(s^k\right)\right\}\right) = \arg\max_i \Psi_i \quad i = 1, \ldots, L\,, \tag{6.11}$$

La definición del elemento de combinación permite enunciar la siguiente afirmación.

Proposición 6.4. *Un clasificador* F *de arquitectura* {*1,1-v-r SVMC, combinación*} *con todos los nodos de dicotomía correctos sobre una entrada* $\mathbf{x} \in \theta_r$*, cumple que* $\Psi\left(\left\{\Theta\left(s^k\right)\right\}\right) = \Psi_r = K - 1$ *y* $\Psi_i = -K + 2\,\forall i \neq r$*.*

Demostración. Una arquitectura {1,1-v-r SVMC, combinación} posee $K-1$ nodos de dicotomía donde la clase θ_r está en solitario por lo que Ψ_r recibe $K-1$ positivos. En cambio no recibe votos negativos puesto que si θ_r provoca una salida nula, el intérprete asigna valores negativos a las clases que están en solitario.

Las clases $\theta_{i\neq r}$ sólo reciben votos negativos ya que si compiten en forma única con θ_r, esta clase gana y no reciben votos, si acompañan a θ_r en la clase 0 no reciben votos, y si compiten en forma única con θ_r en la clase 0, reciben votos negativos. De las $K-1$ veces que está $\theta_{i\neq r}$ en solitario, sólo una vez tiene a θ_r como clase única, el resto de veces, $K-2$, θ_r esta en la clase 0, por lo que $\Psi_i = -K + 2\,\forall i \neq r$. □

Ahora que se ha determinado el estado en que se encuentran todos los cómputos Ψ_r cuando todos los nodos emiten respuesta correcta, se puede estudiar cuál es el factor de robustez ε_{rob} de este tipo de arquitectura. Para determinarlo se hará uso de la siguiente afirmación.

Lema 6.5. *Sea* $d_{ij}^F = \left|\Psi_i^F - \Psi_j^F\right|$ *la distancia entre dos cómputos de votos para un clasificador* F *de arquitectura* {*1,1-v-r SVMC, combinación*}*. Si se produce un nuevo error,* ϵ*, en alguno de los nodos de dicotomía de* F*, entonces se genera un nuevo clasificador* G*, cumpliendo*

$$\max_{\epsilon} d_{ij}^F - d_{ij}^G = 2\,. \tag{6.12}$$

Demostración. Si se observa la definición del intérprete (6.8), un error implicando las clases θ_i y θ_j, cómo máximo puede provocar que una de las clases deje de sumar un voto y en cambio pase a restar uno, o bien que una de las clases deje de sumar un voto y la otra clase implicada sume uno, por lo que en cualquier caso su distancia de votos se reduce a lo sumo en 2 unidades. □

Proposición 6.6. *Una arquitectura de multiclasificación {1,1-v-r SVMC,combinación} posee un factor de robustez*

$$\varepsilon_{rob} = \frac{K-2}{\binom{K}{2}} = \frac{2\,(K-2)}{K\,(K-1)}\,. \tag{6.13}$$

Demostración. Siguiendo la definición de factor de robustez (4.39) de la página 79, puesto que el número de nodos de dicotomía de un esquema de descomposición 1,1-v-r SVMC implicados en cualquier entrada $\mathbf{x}$ de la clase θ_r es $\binom{K}{2}$, tan solo es necesario demostrar que el número mínimo de nodos equivocados que provocan una clasificación final incorrecta es $\#f_r^{err} = K-2$.

Por la proposición anterior se sabe que una arquitectura de multiclasificación {1,1-v-r SVMC, combinación} F con todos sus nodos de dicotomía correctos cumple $\Psi_r = K-1$ y $\Psi_i = -K+2\,\forall i \neq r$, por lo que usando la definición de distancia entre votaciones del lema adjunto puede afirmarse que $d_{ir}^F = \left|\Psi_i^F - \Psi_r^F\right| = 2K-3\,\forall i \neq r$. Finalmente, la ocurrencia de cualquier error, ϵ, provocará que la nueva máquina, G, tenga a la sumo una reducción en 2 unidades en alguna de las distancias entre votaciones anteriores

$$\exists i \neq r \quad : \quad d_{ir}^G = d_{ir}^F - 2\,, \tag{6.14}$$

por lo que el número mínimo de equivocaciones que deben ocurrir en los nodos para que la clasificación final sea equivocada es de

$$\#f_r^{err} = \left\lfloor \frac{2K-3}{2} \right\rfloor = K-2\,, \tag{6.15}$$

donde $\lfloor\cdot\rfloor$ indica la función parte entera, quedando así demostrada la proposición. □

En la Figura 6.1 puede observarse una gráfica que relaciona el número de clases del problema de multiclasificación respecto a la proporción, en tanto por ciento, de nodos de dicotomía cuya salida errónea es admisible sin que exista equivocación en la clasificación final. Esta relación tiende de forma asintótica a 0, pero mantiene valores ciertamente elevados para un número de clases no despreciable. Así, un problema de clasificación de 5 clases implicando 10 nodos de dicotomía permite un porcentaje de error del 30% — hasta 3 nodos equivocados — sin que exista error en la salida final, mientras que un problema con 10 clases permitiría un error del 17.78%, o sea 8 nodos sobre 45.

6.3 Comparativa con la Metodología ECOC

Una tarea que se ha ido relegando hasta el momento es aquella de comparar de la forma más analítica posible la robustez del nuevo esquema de multiclasificación

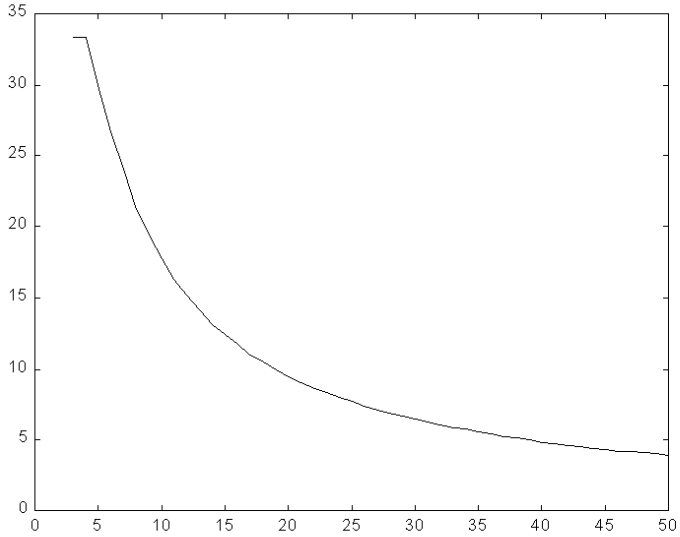

Figura 6.1: Relación entre el número de clases implicadas en la clasificación y el porcentaje de nodos de dicotomía que admiten error en su clasificación sin que el resultado final correcto se vea alterado utilizando máquinas K-SVCR.

basado en el algoritmo K-SVCR respecto a la que se puede obtener mediante una combinación de las predicciones de los nodos de dicotomía mediante la metodología ECOC [Dietterich and Bakiri, 1995]. Como ya fue mostrado en el Capítulo 4, esta técnica, en su formulación estándar, establece una codificación binaria para cada una de las clases que componen el espacio de salida de forma que se intenta maximizar la distancia Hamming entre los diferentes códigos construidos. Un ejemplo para el caso $K = 4$ se describió en (4.8), donde las filas denotan la salida esperada para cada nodo de dicotomía f_k y las columnas representan la codificación de cada clase $\theta_i \in \mathcal{Y}$. El elemento de combinación, Ψ_{Ham}, asigna el ejemplo presentado a aquella clase que posee una codificación más cercana en distancia Hamming a la codificación generada por la entrada.

La codificación de un problema multiclase es muy variada, pudiendo utilizarse diferente cantidad de bits — filas en la matriz $\mathbf{D}_{ECOC}$ — y una gran diversidad de códigos. Siguiendo [Dietterich and Bakiri, 1995], una buena codificación ECOC para un problema multiclase debería de satisfacer dos propiedades:

- Separación de Columnas. Cada código debería estar bien separado en la distancia Hamming de cada uno de los otros códigos.
- Separación de Filas. Cada posición bit de un nodo de dicotomía f_i debería estar no correlacionado con los nodos a ser aprendidos para las otras posiciones bit f_j, $j \neq i$. Esto puede ser conseguido maximizando la distancia Hamming entre filas y entre las complementarias de las filas.

En general, si existen K clases, es posible utilizar a lo sumo $2^{K-1}-1$ filas tras eliminar todas aquellas que son complementarias y las que no emiten juicio cuando todos los valores en la fila son $+1$ o -1.

6.3.1 Robustez de la multiclasificación {Biclase, ECOC}

Encontrar un único método que permita construir adecuadas codificaciones ECOC para cualquier valor de K es todavía un problema de investigación abierto. Un resultado general sobre la robustez de una codificación es el siguiente.

Lema 6.7. *Sea d la distancia de Hamming mínima entre dos códigos pertenecientes a un esquema F de multiclasificación ECOC de L bits — dos columnas de la matriz de descomposición $\mathbf{D}_{ECOC}$, con L nodos de dicotomía — , entonces*

$$\varepsilon_{rob}(F) = \frac{\left\lfloor \frac{d-1}{2} \right\rfloor}{L}. \tag{6.16}$$

Los autores en [Dietterich and Bakiri, 1995] sugieren cuatro métodos diferentes de creación de códigos en función del número de clases tratadas:

1. una técnica exhaustiva,
2. un método que selecciona las filas a partir de una codificación exhaustiva,
3. un método basado en un algoritmo de ascensión aleatorio,
4. códigos BCH.

Técnica Exhaustiva

Cuando $3 \leq K \leq 7$, es posible construir una codificación de longitud $2^{K-1}-1$ cumpliendo que la distancia de Hamming mínima entre dos códigos es 2^{K-2}, por lo que puede enunciarse la siguiente afirmación,

Proposición 6.8. *Una arquitectura de multiclasificación basada en la metodología ECOC creada mediante la técnica exhaustiva posee un factor de robustez,*

$$\varepsilon_{rob} = \frac{2^{K-3}-1}{2^{K-1}-1}. \tag{6.17}$$

Demostración. Basta aplicar el lema anterior, observando que la técnica exhaustiva obliga a una distancia Hamming igual entre todas las codificaciones, $d = 2^{K-2}$, sobre una codificación que ha sido establecida en $2^{K-1}-1$ bits. □

K	L_{K-SVCR}	L_{ECOC}	$\varepsilon_{rob}^{K-SVCR}$	ε_{rob}^{ECOC}
3	3	3	0.333	0.000
4	6	7	0.333	0.143
5	10	15	0.300	0.200
6	15	31	0.267	0.226
7	21	63	0.238	0.238

Tabla 6.3: Número de nodos de dicotomía y factor de robustez para el caso $3 \leq K \leq 7$.

A diferencia de la arquitectura multiclase {1,1-v-r SVMC, combinación}, cuyo factor de robustez tiende asintóticamente a 0 cuando aumenta el número de clases, el factor de robustez de esta arquitectura de multiclasificación cumple

$$\lim_{K\to\infty} \varepsilon_{rob} = \lim_{K\to\infty} \frac{2^{K-3}-1}{2^{K-1}-1} = \frac{1}{4}\,, \tag{6.18}$$

por lo que a nivel general es más robusta que la propuesta basada en K-SVCRs. Sin embargo deben puntualizarse dos hechos a tener en cuenta:

- El número de nodos de dicotomía generados por una arquitectura ECOC mediante la técnica exhaustiva es $2^{K-1}-1$, una cantidad siempre superior, excepto la igualdad para el caso extremo de $K = 3$, a la necesaria para el caso basado en K-SVCRs, como puede observarse en la Tabla 6.3.
- En el caso $3 \leq K \leq 7$, se cumple que $\varepsilon_{rob}^{K-SVCR} \geq \varepsilon_{rob}^{ECOC}$, tal como se aprecia en la misma Tabla 6.3.

Por tanto, puede afirmarse que si el número de clases es pequeño, $3 \leq K \leq 7$, la nueva metodología {1,1-v-r SVMC, combinación}

1. Crea arquitecturas de multiclasificación más robustas que aquellas creadas con el algoritmo ECOC mediante la técnica exhaustiva.
2. El número de nodos de dicotomía es inferior, o igual en el caso extremo $K = 3$.

Selección de Filas a partir de Códigos Exhaustivos

Cuando $8 \leq K \leq 11$, se construye un código exhaustivo y se selecciona un subconjunto adecuado de sus filas aplicando el algoritmo de búsqueda local GSAT. Se requerirá que la solución incluya exactamente L filas — la longitud de código deseada — mientras se asegura que la distancia Hamming entre cualquier par de filas está entre d y $L-d$, para algún valor elegido de d.

K	L	$\varepsilon_{rob}^{K-SVCR}$	ε_{rob}^{ECOC}
8	28	0.214	0.214
9	36	0.194	0.194
10	45	0.178	0.222
11	55	0.164	0.218

Tabla 6.4: Número de nodos de dicotomía y factor de robustez para el caso $8 \leq K \leq 11$.

Para realizar la comparativa con la arquitectura basada en K-SVCRs se ha desarrollado un estudio empírico generando de forma aleatoria codificaciones ECOC con un número de bits igual a aquel que utiliza la arquitectura propuesta sobre las K-SVCRs. Al respecto, debe apuntarse que:

1. Debido a que la finalidad del estudio es comparar la robustez de la multiclasificación, se ha relajado la restricción de maximizar la distancia Hamming entre filas y sólo se ha considerado la maximización de la distancia entre columnas, aunque ello pueda provocar una mayor correlación entre los nodos de dicotomía.

2. La dimensión del espacio de búsqueda de la codificación más robusta aumenta de una forma exponencial respecto al número de clases tratadas; por ejemplo, cuando $K = 9$ las posibles formas de escoger los $L = \binom{K}{2} = 36$ nodos de dicotomía es del orden de 10^{43}, por lo que aunque el número de codificaciones investigado está en un rango del orden de $10^6 - 10^9$, las afirmaciones que puedan realizarse, aún cercanas a la realidad, no son de ningún modo exactas.

Tras realizar las codificaciones, se ha obtenido el resultado que puede apreciarse en la Tabla 6.4 por lo que puede afirmarse, de forma empírica, que la tecnología ECOC resulta más robusta que aquella que se propone sobre máquinas K-SVCR sólo si el número de clases es $K = 10$ u 11, aunque para ello es necesario realizar una búsqueda de la codificación adecuada.

Algoritmo de Ascensión Aleatorio o Códigos BCH

Cuando $K > 11$, es posible realizar un algoritmo de búsqueda aleatorio que alcance un máximo local para las distancias Hamming entre pares de filas y de columnas, o bien utilizar una clase amplia de códigos cíclicos de corrección de error aleatorios conocidos como BCH que emplea métodos algebraicos de la teoría de Galois para diseñar codificaciones quasi-óptimas. La opción de generar de nuevo codificaciones aleatorias para realizar la comparativa podría emplearse, pero los resultados obtenidos no permitirían extraer ninguna conclusión estadísticamente aceptable, por lo que se

K	L	$\varepsilon_{rob}^{K-SVCR}$	ε_{rob}^{ECOC}
12	66	0.151	0.212

Tabla 6.5: Número de nodos de dicotomía y factor de robustez para el caso $K = 12$.

ha optado por realizar tan sólo una búsqueda para el caso $K = 12$ que sirva a modo de ejemplo[1] , obteniéndose el resultado que aparece en la Tabla 6.5.

6.3.2 Máquina ECOC *K*-SVCR

El uso de la teoría ECOC en la combinación de predicciones de los nodos de dicotomía se ha mostrado como una técnica eficiente para lograr aumentar la robustez de una arquitectura de multiclasificación. En el presente apartado se propone la combinación de esta teoría junto a un esquema de descomposición en paralelo formado por nodos de dicotomía del tipo K-SVCR. Se ha de hacer notar sin embargo que si bien se conseguirá una mayor robustez, por contra:

- Ya no será siempre posible controlar de forma directa la precisión de trabajo sobre las clases que se deseen mediante el parámetro δ del nodo de dicotomía.
- La comparación con las técnicas propuestas en [Dietterich and Bakiri, 1995] y algunas modificaciones diseñadas en la presente línea de investigación no será posible realizarla de forma analítica más que en unos pocos casos, por lo que se habrá de recurrir a extracciones aleatorias para realizar el estudio empírico.

Al tratarse de salidas ternarias aquellas que deben ser combinadas, se hace necesario establecer un par de definiciones generalizadoras [MacWillians and Sloane, 1993].

Definición 6.9. *Sean* $\mathbf{x} = (x_1 x_2 \cdots x_L)^\top$ *e* $\mathbf{y} = (y_1 y_2 \cdots y_L)^\top$ *dos vectores de una codificación* $\mathcal{C}$ *de longitud* L *y dimensión* K*. Se define la* distancia Hamming *entre los dos vectores como el número de posiciones en las que difieren, y es denotado por* $dist\,(\mathbf{x}, \mathbf{y})$*.*

Definición 6.10. *Sea* $\mathbf{x} = (x_1 x_2 \cdots x_L)^\top$ *un vector de una codificación* $\mathcal{C}$ *de longitud* L *y dimensión* K*. Se define el* peso Hamming *del vector como el número de posiciones no nulas y es denotado por* $wt\,(\mathbf{x})$*.*

Estas definiciones permiten definir la *distancia mínima de una codificación* como

$$d = \min dist\,(\mathbf{x}, \mathbf{y}) = \min wt\,(\mathbf{x} - \mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{C}\,. \tag{6.19}$$

[1] En este caso el número de combinaciones posible alcanza las $2,18 \cdot 10^{125}$

Elemento de Combinación ECOC

Definición 6.11. *Sea* $\mathbf{s} = \left\{\Theta\left(s^k\right)\right\}_{k=1}^{L} = \left\{Id\left(s^k\right)\right\}_{k=1}^{L} = \left(s^1 s^2 \cdots s^L\right)^\top$ *el conjunto de interpretaciones de los* $L = \binom{K}{2}$ *nodos de dicotomía 1,1-v-r SVMC para una cierta entrada. Si* $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{K} = \left\{\left(c_i^1 c_i^2 \cdots c_i^L\right)^\top\right\}_{i=1}^{K}$ *es la codificación realizada sobre el conjunto de clases, representando cada vector columna* $\mathbf{c}_r = \left(c_r^1 c_r^2 \cdots c_r^L\right)^\top$ *la palabra código de la clase* θ_r*, entonces se define como* elemento de combinación ECOC de la arquitectura 1,1-v-r SVMC *la función*

$$\Psi\left(\left\{\Theta\left(s^k\right)\right\}\right) = \theta_r \quad : \quad dist\left(\mathbf{s}, \mathbf{c}_r\right) = \min_{i=1,\ldots,K} dist\left(\mathbf{s}, \mathbf{c}_i\right) . \tag{6.20}$$

Este tipo de decodificación recibe el nombre de vecino más próximo.

6.3.3 Comparativa ECOC *K*-SVCR *versus* {Biclase, ECOC}

Se intentará establecer, en función del factor de robustez, qué tipo de arquitectura multiclase resulta más adecuado para realizar la clasificación. Para llevar a cabo el estudio comparativo se han tenido en cuenta los siguientes tipos de máquinas:

- Arquitectura {1,1-v-r SVMC, combinación} basada en $L = \binom{K}{2}$ nodos de dicotomía ternarios K-SVCR. Se denotará como $A3^{combi}$. Correspondería a la arquitectura desarrollada hasta el momento.
- Arquitectura {{paralelo, árbol} binario, ECOC} basada en $L = 2^{K-1} - 1$ nodos de dicotomía binarios en formato paralelo o arbóreo. Se denotará como $A2_{std}^{ECOC}$. Corresponde a la arquitectura también notada como {Biclase, ECOC}.
- Arquitectura {{paralelo, árbol} binario, ECOC} basada en un número reducido, $L = \binom{K}{2}$, de nodos de dicotomía binarios en formato paralelo o arbóreo. Se denotará como $A2_{red}^{ECOC}$. Se trata de una arquitectura {Biclase, ECOC} desarrollada sobre un número pequeño de nodos de dicotomía.
- Arquitectura {{paralelo, árbol} binario, ECOC} basada en un subconjunto de códigos elegidos entre aquellos de igual peso Hamming, $wt\left(\mathbf{x}\right)$, de forma que se obtenga el mayor factor de robustez posible. Esto se consigue cuando la distancia entre las palabras códigos es lo más igual posible e igual a la distancia mínima [Bakiri and Dietterich, 2000]. Se denotará como $A2_{d=}^{ECOC}$.
- Arquitectura {1,1-v-r SVMC, ECOC} obtenida a partir de la anterior mediante la inclusión del tercer elemento — etiqueta 0 — de forma aleatoria sobre las palabras código en una cantidad del orden de $K/3$ sobre cada vector fila de la matriz de codificación o descomposición $\mathbf{D}$. Representa la máquina ECOC

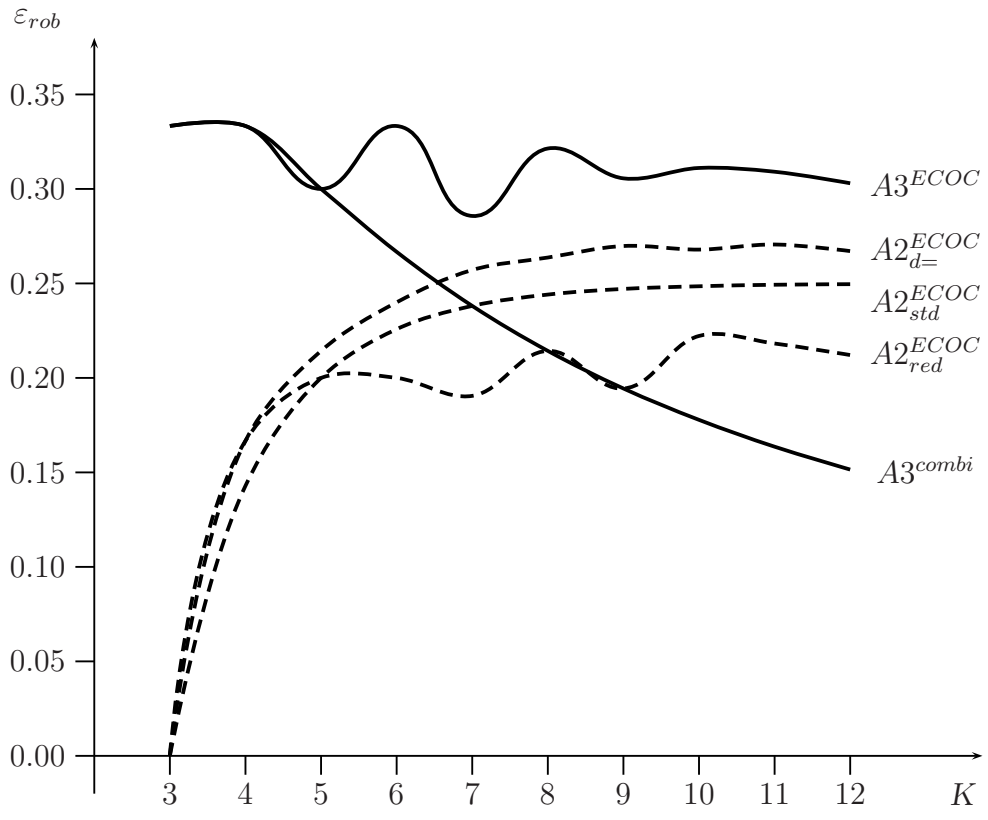

Figura 6.2: Factor de robustez de las diferentes arquitecturas empleadas en la comparativa en función del número de clases K.

> K-SVCR, que debería recoger las buenas características de robustez aportadas tanto por la tecnología ECOC como por el hecho de utilizar clasificadores ternarios. También se denotará como $A3^{ECOC}$.

En función del número K de clases que intervienen en el problema de clasificación se ha obtenido la gráfica de la Figura 6.2 donde se representa el factor de robustez de cada una de las arquitecturas, hallado en la mayoría de las ocasiones de forma empírica. De igual forma, se presentan en la Tabla 6.6 y la Tabla 6.7 los datos más relevantes correspondientes a las pruebas realizadas para elaborar la comparación.

Caso $K = 3$ ó 4

Cuando el número de clases es tan reducido, la arquitectura propuesta inicialmente, $A3^{combi}$, posee el mayor factor de robustez, siendo equivalente a aquel obtenido por la arquitectura más sofisticada, $A3^{ECOC}$. Las máquinas basadas en nodos de dicotomía binarios poseen un factor ε_{rob} muy pequeño. Esta comparativa, al ser el número de

		$A3^{combi}$			$A2^{ECOC}_{red}$			$A3^{ECOC}$		
K	L	d_{min}	d_{max}	ε_{rob}	d_{min}	d_{max}	ε_{rob}	d_{min}	d_{max}	ε_{rob}
3	3			0.333	2	2	0.000	3	3	0.333
4	6			0.333	3	4	0.167	5	5	0.333
5	10			0.300	6	6	0.200	7	8	0.300
6	15			0.267	8	8	0.200	11	13	0.333
7	21			0.238	10	12	0.190	14	17	0.286
8	28			0.214	13	17	0.214	19	23	0.321
9	36			0.194	16	21	0.194	24	28	0.306
10	45			0.178	21	28	0.222	30	35	0.311
11	55			0.164	26	35	0.218	36	46	0.309
12	66			0.151	30	45	0.212	42	51	0.303

Tabla 6.6: Relación de resultados sobre igual número de nodos de dicotomía, L.

	$A2^{ECOC}_{std}$				$A2^{ECOC}_{d=}$			
K	L	d_{min}	d_{max}	ε_{rob}	L	d_{min}	d_{max}	ε_{rob}
3	3	2	2	0.000	3	2	2	0.000
4	7	4	4	0.143	6	3	4	0.167
5	15	8	8	0.200	14	7	8	0.214
6	31	16	16	0.226	25	14	14	0.240
7	63	32	32	0.238	35	20	20	0.257
8	127	64	64	0.244	91	50	50	0.264
9	255	128	128	0.247	126	70	70	0.270
10	511	256	256	0.249	336	182	182	0.268
11	1023	512	512	0.249	462	252	252	0.271
12	2047	1024	1024	0.250	1254	672	672	0.267

Tabla 6.7: Relación de resultados sobre igual distancia entre códigos, d.

clases pequeño ha podido ser llevada a cabo de forma analítica, estudiando todas las posibles combinaciones.

Caso $K = 5$ ó 6

El número de clases puede considerarse todavía pequeño, pero una gran parte de los problemas de clasificación extraídos de la realidad se corresponden con esta característica. Puede observarse de la Figura 6.2 o de las Tablas 6.6 y 6.7 asociadas que las arquitecturas basadas en clasificadores ternarios vuelven a comportarse con una mayor robustez. De nuevo, el estudio ha sido elaborado sobre todas las posibles combinaciones de bits en los clasificadores binarios, por lo que puede afirmarse,

Proposición 6.12. *En un problema de clasificación multiclase, si el número de clases es menor o igual que* 6*, $K \leq 6$, entonces las arquitecturas $A3^{combi}$ y $A3^{ECOC}$, basadas en clasificadores ternarios K-SVCR, poseen un factor de robustez mayor que las arquitecturas basadas en clasificadores binarios $A2^{ECOC}_{std}$, $A2^{ECOC}_{red}$ y $A2^{ECOC}_{d=}$.*

La elección final de la arquitectura ternaria a emplear será un compromiso establecido por el usuario entre el deseo de robustez y el tiempo de cálculo necesario para determinar las clases a ser clasificadas por cada nodo de dicotomía.

Caso $K \in [7, 9]$

En esta ocasión se realizará un paralelismo entre las arquitecturas construidas sobre $L = \binom{K}{2}$ nodos de dicotomías y aquellas desarrolladas sobre una cantidad mayor de nodos, cuyo número se notará como L'. El motivo reside en el espectacular incremento del número de clasificadores a realizar en el caso de utilizar las arquitecturas $A2^{ECOC}_{std}$ o $A2^{ECOC}_{d=}$ para obtener una mejora en el factor ε_{rob}. De entre las arquitecturas con L nodos de dicotomía vuelven a obtener un mayor factor de robustez aquellas basadas en clasificadores ternarios, si bien la $A3^{combi}$ no necesita de proceso de cálculo para determinar las clases implicadas en cada nodo.

Las arquitecturas basadas en L' nodos superan en robustez a $A2^{ECOC}_{red}$ y $A3^{combi}$, aunque para ello utilizan un número mucho mayor de nodos. Aprovechando el cálculo ECOC realizado para estas arquitecturas, la $A3^{ECOC}$ obtendrá mayor robustez que las anteriores con sólo un pequeño aumento de tiempo de computación inicial y se podrá reducir de forma considerable el número de nodos de dicotomía.

Caso $K \geq 10$

El uso de la arquitectura $A3^{combi}$ propuesta inicialmente deja de ser interesante desde el punto de vista de robustez si se compara con el resto de posibilidades, si bien necesita de un proceso de cálculo menor. De entre las arquitecturas basadas en clasificadores biclase, parece resultar más adecuada, al menos de forma empírica, la $A2^{ECOC}_{d=}$, si bien necesita de un número de nodos de dicotomía L' mucho mayor que $A2^{ECOC}_{red}$, aunque menor que $A2^{ECOC}_{std}$. De nuevo será el usuario quien deba determinar el equilibrio entre tiempo de cálculo de clases a separar por cada nodo — que aumentará de forma exponencial con el número de clases —, el factor de robustez y el número de nodos de dicotomía.

Sin duda resulta la mejor opción, al menos de forma empírica, utilizar la arquitectura $A3^{ECOC}$: la expansión se realiza sobre sólo L nodos de dicotomía, permite aprovechar el tiempo de cómputo utilizado para aplicar la teoría ECOC a separadores binarios, y, desde luego, posee el mayor factor de robustez de entre todas las arquitecturas analizadas.

6.4 Experimentación

La propiedad de robustez de la nueva estructura basada en la máquina K-SVCR en entornos de multiclasificación ha sido demostrada y su evaluación se ha realizado hasta el momento sobre experimentos artificiales estándares en la literatura. Para acabar de comprobar su comportamiento satisfactorio se han escogido algunas bases de datos o problemas tipo que usualmente son utilizados por la comunidad de 'Machine Learning'. Para tal fin se han utilizado conjuntos de datos estándares como son los del 'UCI Repository' [Blake and Merz, 1998], en concreto se analizan las prestaciones de la nueva arquitectura sobre los bancos de datos 'Iris', 'Glass' y 'Wine'. Esta elección se ha realizado para así poder comparar los resultados obtenidos con aquellos presentados en el trabajo de [Weston and Watkins, 1998].

Si hasta el momento todas las experimentaciones han sido realizadas utilizando la formulación teórica original de Vapnik de las SVMs, el hecho de experimentar sobre conjuntos de datos de tamaño mediano provoca que la resolución del problema de programación cuadrática consuma un tiempo computacional prohibitivo sobre computadores tipo PC. Es este problema, el del tiempo de computación, aquel que más ha frenado la estandarización de las SVMs como máquinas de aprendizaje en la comunidad científica, por lo que pudiera decirse que es el foco de investigación principal de los desarrolladores de SVMs. No se tiene la intención de detallar y evaluar ahora todos los posibles métodos para acelerar el aprendizaje de las SVMs — casi todos ellos basados en técnicas de particiones en subconjuntos o 'chunking' [Platt, 1999c], [Platt, 1999a], [Keerthi et al., 1999] — ni de entrar a comentar el programario de

	$\#pts$	$\#atr$	$\#clase$	1-v-r	1-v-1	qp-mc-sv	K-SVCR
iris	150	3	4	1.33	1.33	1.33	$[2.93, 4.0]$
wine	178	13	3	5.6	5.6	3.6	$[2.29, 4.29]$
glass	214	9	7	35.2	36.4	35.6	$[35.47, 39.35]$

Tabla 6.8: Prestaciones en porcentaje de error sobre el conjunto de test de la nueva máquina de aprendizaje respecto a combinaciones estándares de máquinas SVM y máquinas SVM multiclase.

desarrollo más apropiado [2] . Por motivos prácticos [3] se optó por utilizar un procedimiento iterativo de mínimos cuadrados con pesos, IRWLS — del inglés, *Iterative Re-Weighted Least Square* — desarrollado por investigadores de la Universidad de Alcalá y de la Universidad Carlos III de Madrid [Pérez-Cruz et al., 2000] que asegura, tal como se ha podido comprobar en el presente estudio, una resolución del problema QP rápida y con resultados idénticos a aquellos obtenidos por la resolución estándar del problema.

Siguiendo el procedimiento establecido en [Weston and Watkins, 1998], cada base de datos fue partida aleatoriamente con una décima parte de los datos siendo usados como conjunto de test, aunque a diferencia del trabajo citado, este proceso se realizó no sólo 10 veces, sino que se repitió para 100 elecciones diferentes con el objetivo de asegurar la validez de los resultados desde un punto de vista estadístico. Los datos fueron preprocesados para normalizarlos a media nula y varianza unitaria y se escogieron diferentes núcleos para cada 'benchmark': núcleos polinomiales de grado 2 y 3 en el caso de los conjuntos 'Iris' y 'Wine', respectivamente, y núcleos función gaussiana para los datos de 'Glass', con $\sigma = 0.50$. Para hacer posible la comparación, todos los algoritmos utilizaron $C = \infty$ y un valor de insensitividad muy pequeño $\delta = 0.07$ puesto que los datos de entrenamiento deben ser clasificados sin error. Teniendo en cuenta estas apreciaciones, en la Tabla 6.8 se recogen los resultados obtenidos con el nuevo procedimiento en comparación con aquellos presentados en el trabajo de referencia, donde también se muestran el número de patrones de entrenamiento, el número de atributos y el número de clases de cada base de datos. Los resultados para la nueva máquina han sido expresados en forma de intervalo con la intención de no deshacer los empates producidos en la clasificación sobre el conjunto de test. La cota inferior del intervalo determina el error que se cometería si cada vez que existe un empate y en él se halla implicada la etiqueta adecuada, la máquina la escogiera. La cota superior en cambio significaría el porcentaje de error cometido sobre el conjunto de test si cada vez que hay un empate la máquina escogiera como salida alguna etiqueta equivocada.

[2] Un buen punto de referencia para hallar software apropiado y cualquier temática relacionada con SVMs en general es la dirección URL: http://www.kernel-machines.org/

[3] Ha de agradecerse a Fernando Pérez Cruz de la Universidad de Alcalá su trabajo adaptando el programario de resolución IRWLS a la nueva máquina de aprendizaje K-SVCR bajo entorno de simulación Matlab.

Como puede observarse, el nuevo procedimiento de multiclasificación se comporta con unos resultados de porcentaje de error similares a aquellos obtenidos por las otras máquinas. Detallando, el peor comportamiento se recoge sobre la evaluación del conjunto 'Iris'. En este caso el error es pequeño en términos absolutos, pero elevado si se relativiza respecto a las otras máquinas. Debe hacerse notar sin embargo que estos resultados dependen en gran medida del número de evaluaciones realizadas. Para intentar imitar el método en [Weston and Watkins, 1998], se realizaron los procesos de simulación de 10 en 10 iteraciones hasta llegar a 100, y se corroboró que en bastantes de estas iteraciones parciales se conseguían niveles de error similares a los de las otras máquinas. Para el caso de la base 'Glass', el porcentaje de error es muy difícil de reducir puesto que existen dos categorías a las que pertenecen la mayoría de los patrones mientras que otras clases poseen muy pocos representantes. Al realizar la elección aleatoria de patrones, algunas clases están muy poco representadas durante el entrenamiento por lo que el test sobre elementos con estas etiquetas dará error en numerosas ocasiones. Una posible mejora para este caso concreto quizás se podría conseguir dando pesos diferentes a los votos en función del número de patrones que están representando a cada clase durante el entrenamiento. En el último de los 'benchmark' escogidos, 'Wine', el número de clases es reducido, 3 clases, y cada una de ellas tiene un número parecido de representantes, 59, 71 y 48, por lo que aunque el espacio de entrada es mayor, 13 atributos, la nueva máquina está mejor dispuesta para realizar una buena clasificación como así se observa en los resultados de la Tabla 6.8.

Finalmente señalar que el empleo de un factor de insensitividad tan reducido provoca que el número de vectores soporte de las máquinas entrenadas sea elevado, pero era esta la única forma de asegurar que la anchura del intervalo de error fuera reducida. A modo de ejemplo, en el caso de la base de datos 'Wine', cuando el factor δ aumenta a 0.50 entonces el intervalo de porcentaje de error viene definido sobre 100 iteraciones como $[1.59, 11.88]$, de forma que el posible máximo error aumenta significativamente, aunque el mínimo error también se ve reducido.

6.5 En Resumen

La técnica empleada por las SVMs de introducir el espacio inicial en otro de mayor dimensión para conseguir clasificadores no lineales ha sido empleada de una forma generalizada para crear una nueva máquina de aprendizaje con el objetivo de multiclasificación.

El algoritmo K-SVCR se presenta, inicialmente para el caso separable y más tarde de forma general, como una nueva máquina ideada para ser utilizada en el esquema de descomposición para reconocimiento de patrones multiclase. Esta máquina construye una función de decisión ternaria mediante una doble partición binaria que permite separar dos clases del resto de las otras clases. Algunas de sus características

de funcionamiento y prestaciones han sido experimentadas e ilustradas sobre datos artificiales en el plano.

Estos clasificadores pueden ser combinados en un esquema estándar en paralelo o mediante una formulación en árbol de decisión. Debido a razones de robustez, han sido implementados esquemas en paralelo y su funcionalidad ha sido comparada con la del resto de metodologías analizadas en el Capítulo anterior.

Una correcta interpretación de las respuestas parciales para definir la respuesta global en el esquema de reconstrucción ha permitido mostrar la tolerancia a fallos que presenta el nuevo algoritmo. Por último, su modo de funcionamiento ha sido testado sobre datos artificiales en $\mathbb{R}^2$ y sobre algunos bien conocidos UCI 'benchmarks'.

Parte II

Aplicaciones