Capítulo 4

Clasificación Multiclase con Máquinas de Soporte Vectorial

"¿Quién decide cuando los médicos no se ponen de acuerdo?"
Anónimo.

Para comenzar

Es muy habitual cuando se trabaja con problemas de clasificación extraídos de la vida real encontrarse con situaciones de multiclasificación. Tradicionalmente, cuando se realiza el desarrollo teórico de una máquina de aprendizaje, si ésta ha sido especialmente diseñada para casos binarios como las SVMCs, se soluciona la posibilidad de trabajo sobre un entorno multiclase afirmando que su generalización a tales problemas es "*evidente*". En otras ocasiones la máquina ya está concebida para el trabajo con múltiples salidas, como en el caso de los MLPs, pero el grueso del desarrollo teórico se reduce al caso binario por motivos de simplicidad de notación, pues la generalización a casos multiclase se hace "*obvia*". Por último, existe un tercer tipo de máquina de aprendizaje, como por ejemplo los árboles de decisión, que trabajan directamente con problemas multiclase aunque para ello necesiten de unos nodos de decisión que son, en la mayor parte de las ocasiones, dicotomías.

En el presente capítulo serán desarrolladas y analizadas las diferentes arquitecturas existentes en la literatura sobre reducción de problemas multiclase en particiones biclase que permiten utilizar o incluir las SVMCs como máquina de aprendizaje en el proceso de descomposición del problema general en dicotomías. Un estudio detallado sobre su modo de funcionamiento mostrará tanto las ventajas de su implementación como las dificultades que evidencian cuando es testada su robustez frente a fallos parciales de predicción, la interpretación que realizan de las respuestas, la identificación y/o resolución de errores de trabajo, ...

Por otra parte, en los últimos años han sido desarrolladas SVMs multiclase considerando todas las clases a la vez durante el proceso de aprendizaje, mediante el uso de diferentes algoritmos. Aunque la comparativa de estas metodologías con la mejora propuesta en el Capítulo 5 de la presente memoria no es de ningún modo directa, sí que será analizado el coste computacional que significa su utilización con el fin de realizar un paralelismo con el nuevo método que se propone.

4.1 Definiendo la Clasificación Multiclase

Definición 4.1. *El* problema general de clasificación multiclase a partir de ejemplos *se define como una particularización del PGAE en el caso que el espacio de salida del sistema y de la máquina de aprendizaje sea un conjunto finito cuyos elementos pueden poseer o no una ordenación,* $y \in \mathcal{Y} = \{\theta_1, \ldots, \theta_{K>2}\}$*, pero en cualquier caso el número de estas etiquetas definitorias de clase es estrictamente mayor que dos.*

Se trata de un problema general que puede ser tanto de reconocimiento de patrones como de regresión ordinal, tal como se definió en el Capítulo 1, con la puntualización de no tratarse de un problema binario.

El problema de gran escala original, $K > 2$, es solucionado habitualmente mediante la combinación de funciones de decisión biclase: un *esquema de descomposición* inicial transforma la K-partición en una serie de L biparticiones, $f_1, \ldots, f_L$, mientras que un *método de reconstrucción* posterior realiza la fusión de las predicciones de los L clasificadores para seleccionar una de las K clases como respuesta final.

Definición 4.2. *Se denomina* máquina de clasificación multiclase *a la arquitectura de máquinas de aprendizaje capaz de responder con una etiqueta de clasificación a cualquier entrada.*

Definición 4.3. *Se denomina* nodo de dicotomía *a cada una de las biparticiones que componen la arquitectura de la máquina de clasificación multiclase y son capaces de generar una predicción o respuesta parcial.*

Siguiendo esta metodología, la arquitectura final del clasificador global, fusión de clasificadores parciales, dependerá de:

- el acondicionamiento, interpretación y/o agrupación de las entradas multiclase para realizar las clasificaciones binarias;
- el tipo o tipos de máquina de aprendizaje elegido para realizar la descomposición del espacio de clasificación en biparticiones;
- el esquema de reconstrucción que realiza la fusión de las predicciones;

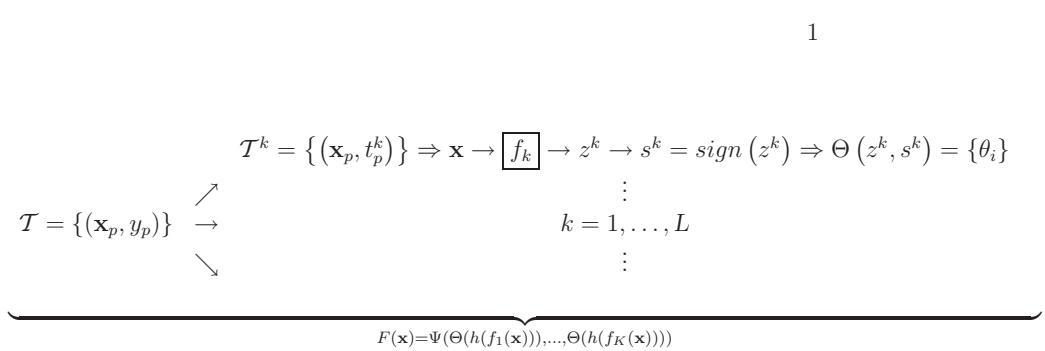

Figura 4.1: Modelo general de una arquitectura de aprendizaje en multiclasificación siguiendo un esquema de descomposición en paralelo.

- la interpretación de la respuesta facilitada por el clasificador global.

Por motivos de claridad se van a establecer las siguientes convenciones de notación. Sea

$$\mathcal{T} = \{(\mathbf{x}_p, y_p)\}_{p=1}^{\ell} \subset \mathcal{X} \times \mathcal{Y} = \mathcal{X} \times \{\theta_1, \ldots, \theta_{K>2}\} \sim P_{\mathcal{X}\mathcal{Y}}^{\ell}, \tag{4.1}$$

el *conjunto de entrenamiento* conteniendo los ℓ pares que constituyen los vectores de entrada y la respuesta del sistema multiclase. Se notará como ℓ_i el número de patrones cuya respuesta es θ_i, $\ell = \sum_{i=1}^{K} \ell_i$. Sin pérdida de generalidad, se supondrá que los patrones están ordenados de forma que los ℓ_1 primeros son de la clase con etiqueta θ_1 y se seguirá este esquema hasta los ℓ_K últimos que serán de la clase con etiqueta θ_K.

Puesto que el problema original va a ser transformado en dicotomías, se hace necesario transformar el conjunto de entrenamiento original $\mathcal{T}$ para cada nodo de dicotomía, por lo que se notará como $t_p^k \in \{-1, 1\}$ la respuesta correcta que debería recibir un vector de entrada $\mathbf{x}_p$ cuando constituye un patrón de entrenamiento de un nodo de dicotomía f_k. Será posible que cada uno de estos clasificadores biclase no trabajen sobre todo el espacio de entrenamiento, por lo que se notará como $\mathcal{T}^k = \mathcal{T}_+^k \cup \mathcal{T}_-^k$ al subconjunto de entrenamiento de la dicotomía f_k compuesto por los elementos a los que se ha dotado de salida positiva o negativa.

Una vez aplicado el nodo de dicotomía f_k al conjunto de entrenamiento modificado $\mathcal{T}^k$, se define como $z^k = f_k(\mathbf{x})$ a la respuesta numérica facilitada por el hiperplano separador construido por el nodo de dicotomía f_k a una entrada $\mathbf{x}$; como $s^k = h(f_k(\mathbf{x})) = sign(z^k)$ al signo que retorna la función de decisión, que para una entrada patrón $\mathbf{x}_p$ debería ser t_p^k; como $\Theta(z^k, s^k) = \theta_i$ a la función que adjudica una etiqueta parcial a esa entrada y, por último, como $F(\mathbf{x}) = \Psi(\Theta(h(f_1(\mathbf{x}))), \ldots, \Theta(h(f_K(\mathbf{x}))))$ a la respuesta global de la máquina de clasificación multiclase.

Si se define como L_r al conjunto de índices correspondientes a los patrones de la clase θ_r, $\#L_r = \ell_r$, se podrá definir más fácilmente el subconjunto de entrenamiento

modificado para entrenar un nodo de dicotomía f_k con los patrones que pertenecen a la clase r como $\mathcal{T}_r = \left\{\left(\mathbf{x}_p, t_p^k\right) \in \mathcal{X} \times \{-1, 1\} : p \in L_r\right\}$, lo que permite definir el conjunto completo de entrenamiento modificado como $\mathcal{T}^* = \bigcup_{r=1}^{K} \mathcal{T}_r$.

4.2 Arquitectura de Descomposición y Reconstrucción

4.2.1 Arquitecturas Estándares

Esquemas de Descomposición

Definición 4.4. *Un k-ésimo nodo de dicotomía dentro de una arquitectura de máquina de clasificación multiclase se denomina* 1-v-r SVMC *— del inglés, one-versus-rest — si es una SVMC entrenada con etiquetas positivas $t_p^k = 1$ para los ℓ_k patrones de entrenamiento de la k-ésima clase etiquetada θ_k, $\mathcal{T}_+^k = \mathcal{T}_k$, y con etiquetas negativas $t_p^k = -1$ para los $\ell - \ell_k$ patrones de las $K-1$ clases restantes, $\mathcal{T}_-^k = \mathcal{T}^* \backslash \mathcal{T}_k$.*

Un 1-v-r clasificador es entrenado para separar una clase de las $K-1$ restantes. El método estándar de descomposición de un problema general de clasificación multiclase a partir de ejemplos en dicotomías consiste en situar $L = K$ clasificadores binarios, tantos como clases, del estilo 1-v-r en paralelo. En el caso de las SVMCs la arquitectura fue propuesta originalmente por [Vapnik, 1995], [Cortes and Vapnik, 1995].

Puede observarse que el tiempo de entrenamiento de este método estándar es linealmente proporcional al número de clases, K. Todos los nodos de dicotomía son entrenados sobre el conjunto de entrenamiento completo, $\mathcal{T}^k = \bigcup_{r=1}^{K} \mathcal{T}_r \quad \forall k = 1, \ldots, K$, con el consiguiente costo computacional, pero también con la mejora que supone contar con toda la información en la fase de entrenamiento de cada nodo. Finalmente destacar que desde el punto de vista teórico deja de existir una cota del error de generalización para la máquina de clasificación multiclase global.

Definición 4.5. *Un k-ésimo nodo de dicotomía, dentro de una arquitectura de máquina de clasificación multiclase se denomina* 1-v-1 SVMC *— del inglés, one-versus-one — si es una SVMC entrenada con etiquetas positivas $t_p^k = 1$ para los ℓ_i patrones de entrenamiento de la i-ésima clase etiquetada θ_i, $\mathcal{T}_+^k = \mathcal{T}_i$, y con etiquetas negativas $t_p^k = -1$ para los ℓ_j patrones de entrenamiento de la j-ésima clase etiquetada θ_j, $\mathcal{T}_-^k = \mathcal{T}_j$.*

El método de descomposición asociado a este tipo de dicotomías consiste en situar $L = K\left(K-1\right)/2$ clasificadores binarios del estilo 1-v-1 en paralelo, cada nodo siendo

entrenado por sólo dos de las K clases implicadas en la clasificación múltiple, $\mathcal{T}^k = \mathcal{T}_i \cup \mathcal{T}_j$ [Hastie and Tibshirani, 1998].

A pesar que algunos autores defienden que el esquema asociado al 1-v-1 SVMC es, en general, preferible al esquema 1-v-r [Kressel, 1999], como así muestran sus estudios empíricos, existen tres inconvenientes asociados al método de descomposición 1-v-1 que ponen en entredicho esta afirmación:

- La 1-v-1 SVMC sólo es entrenada sobre un subconjunto modificado de $\mathcal{T}$, por lo que la fisonomía que adopte la función decisión no tendrá en consideración la existencia de otros patrones de entrenamiento pertenecientes a clases distintas.
- Si la respuesta final dada por la máquina de clasificación multiclase global F es por ejemplo θ_b, todas las predicciones dadas por los (i,j)-ésimos [1] nodos de dicotomía, con $i, j \neq b$, no deberían ser tomadas en cuenta pues la respuesta a patrones de la clase θ_b no pertenece a su tarea de clasificación.
- El tamaño de la máquina de clasificación multiclase asociada a nodos de dicotomía 1-v-1 puede crecer superlinealmente con K.

Mientras que el tercer ítem es una cuestión de tiempo computacional que no debe alterar en principio la bondad de la respuesta final, los dos primeros inconvenientes están directamente relacionados con el hecho de que el nodo de dicotomía no ha tomado en consideración todo el espacio de trabajo. Sería oportuno construir una máquina que capture las propiedades positivas del esquema 1-v-1 que lo hacen empíricamente preferible al 1-v-r y que por otro lado cada clasificador sea entrenado sobre todo el conjunto de aprendizaje, $\mathcal{T}^k = \mathcal{T}^*$, para que sus respuestas sean consecuentes sea cual sea la entrada al nodo de dicotomía. Una forma de unificar ambos esquemas de descomposición dentro de un marco común más general es posible si se especifican mediante una matriz de descomposición [Moreira and Mayoraz, 1998].

Definición 4.6. *Se denomina* matriz de descomposición $\mathbf{D} \in \{-1, 0, +1\}^{L \times K}$ *a aquella que tiene por componentes*

$$D_{ki} = \begin{cases} +1 & si\ \mathcal{T}_i \subset \mathcal{T}_+^k \\ -1 & si\ \mathcal{T}_i \subset \mathcal{T}_-^k \\ 0 & si\ \mathcal{T}_i \cap \mathcal{T}^k = \emptyset \end{cases} . \tag{4.2}$$

La validez de un esquema de descomposición se expresa mediante la restricción que para cualquier par de columnas de $\mathbf{D}$, debe existir al menos una fila para la cual los coeficientes en las dos columnas son $+1$ y -1. Por ejemplo, una máquina de

[1] Con esta notación se pretende resaltar que el k-ésimo nodo de dicotomía es entrenado sólo sobre $\mathcal{T}_i \cup \mathcal{T}_j$.

clasificación multiclase, $K = 4$, con (a) 1-v-r nodos de dicotomía, o (b) 1-v-1 nodos de dicotomía, tiene por matrices de descomposición

$$\mathbf{D}_{1-v-r} = \begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix} \quad , \quad \mathbf{D}_{1-v-1} = \begin{pmatrix} +1 & -1 & 0 & 0 \\ +1 & 0 & -1 & 0 \\ +1 & 0 & 0 & -1 \\ 0 & +1 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & -1 \end{pmatrix} \tag{4.3}$$

(a) (b)

Métodos de Reconstrucción

Durante el esquema de descomposición estándar se construyen L nodos de dicotomía, f_k, en paralelo que son entrenados sobre modificaciones del conjunto de aprendizaje, $\mathcal{T}_k = \left\{\left(\mathbf{x}_p, t_p^k\right) \in \mathcal{X} \times \{-1, +1\}\right\}$, de forma que de acuerdo a la matriz de descomposición los elementos de unas clases son asignados a salidas positivas, los de otras a salidas negativas y los patrones restantes no son tenidos en consideración en aquel clasificador en particular.

Cada nodo de dicotomía f_k entrenado emite una respuesta en forma numérica $z^k = f_k(\mathbf{x})$ a una entrada $\mathbf{x}$. La información más importante en esta respuesta, en principio, se encuentra en el signo $s^k = h(f_k(\mathbf{x})) = sign\left(z^k\right)$ que adopta la función de decisión. En la determinación de la respuesta final facilitada por el método de reconstrucción de la máquina de aprendizaje multiclase han de ser tomados en consideración los siguientes elementos:

- Las *predicciones numéricas parciales* de los nodos de dicotomía, $z^k = f_k(\mathbf{x})$.
- El *signo* de las predicciones numéricas, $s^k = h(f_k(\mathbf{x})) = sign\left(z^k\right)$.
- Un *elemento intérprete* de las predicciones numéricas y binarias, $\Theta\left(z^k, s^k\right)$, con el fin de asignar o no, una o varias clases como posible respuesta de clasificación a una entrada $\mathbf{x}$.
- Un *elemento de combinación* de las predicciones, $\Psi\left(\Theta\left(z^1, s^1\right), \ldots, \Theta\left(z^L, s^L\right)\right)$ que tenga o pueda tener en consideración las predicciones numéricas, sus signos y/o la clase o clases asignadas.

En esta Subsección se realizará un estudio pormenorizado sobre los métodos de reconstrucción más utilizados o estándares y su implicación con el tipo de esquema de descomposición elegido en la primera fase de construcción de la arquitectura de multiclasificación.

Esquemas de Votación Son la forma de reconstrucción más habitual. Se tiene en consideración sólo el signo de las predicciones de todos los nodos de dicotomía. Estos signos son interpretados en función de las clases implicadas en el nodo de dicotomía utilizado en el esquema de descomposición:

- k-ésimo 1-v-r nodo de dicotomía:

$$\Theta\left(s^k\right) = \begin{cases} \theta_k & \text{si } s^k = +1 \\ \emptyset & \text{si } s^k = -1 \end{cases} \quad . \tag{4.4}$$

- (i,j)-ésimo 1-v-1 nodo de dicotomía:

$$\Theta\left(s^k\right) = \begin{cases} \theta_i & \text{si } s^k = +1 \\ \theta_j & \text{si } s^k = -1 \end{cases} \quad . \tag{4.5}$$

Tras la interpretación de las predicciones, el elemento de combinación Ψ realiza un recuento del número de clases votadas, acción de la que toma el nombre el esquema de reconstrucción, que posee diferentes variantes. Adoptando la notación de Ψ_i para indicar el número de votos recibidos por la clase θ_i del elemento intérprete $\Theta\left(s^k\right)$ sobre todos los nodos de dicotomía, f_k, se definen a continuación alguna de estas posibilidades.

Votación por Unanimidad: se determina como respuesta aquella única clase θ_r que haya obtenido todos los votos posibles en las predicciones.

En el caso 1-v-r significa que sólo debe existir un voto válido, $\Psi_r = 1$, siendo los demás nulos, $\Psi_{i\neq r} = 0$. Para el esquema 1-v-1 el número de votos obtenidos por θ_r ha de ser $\Psi_r = L - 1$, que es el número de nodos de dicotomía en los que cualquier clase θ_i está implicada, mientras que es irrelevante el número de votos obtenidos por el resto de clases, $\Psi_{i\neq r} < L - 1$.

Votación por Mayoría Absoluta: se determina como respuesta final aquella única clase θ_r que haya obtenido más de la mitad de los votos posibles.

En la metodología 1-v-r, con el intérprete $\Theta\left(s^k\right)$ definido en (4.4), sólo es posible como máximo un voto por clase ya que cada clase esta implicada con etiqueta positiva en sólo un nodo de dicotomía, por lo que este tipo de votación es equivalente al voto por unanimidad. Para el caso 1-v-1, se ha de cumplir que $\Psi_r > \Psi_{i\neq r}$ y que $\Psi_r \geq (L-1)/2$.

Votación por Mayoría Simple: se determina como respuesta final aquella única clase θ_r que haya obtenido más votos que el resto de clases.

De nuevo en la metodología 1-v-r este esquema de votación es equivalente a los anteriores. En el caso 1-v-1, ahora basta con cumplir que $\Psi_r > \Psi_{i \neq r}$. Este algoritmo de combinación de dicotomías 1-v-1 también recibe el nombre de *Max Wins* [Friedman, 1996].

Variaciones de los Esquemas de Votación sobre dicotomías 1-v-r Si los nodos de dicotomía son entrenados todos sin error sobre el conjunto de entrenamiento, la votación por unanimidad debería bastar como método de reconstrucción para obtener una respuesta adecuada. Esta afirmación, aunque deseable, es falsa, como se ejemplariza en [Kressel, 1999]. En numerosas ocasiones la respuesta final será un empate, una ambigüedad en la respuesta, por lo que la entrada se habrá quedado sin etiquetar. Una forma usual de romper estas situaciones de empate consiste en evaluar el elemento intérprete sobre las predicciones numéricas, además de utilizar su signo. Así, se define una variación de (4.4) como

$$\Theta\left(z^k, s^k\right) = \begin{cases} \left(\theta_k, z^k\right) & \text{si } s^k = +1 \\ \emptyset & \text{si } s^k = -1 \end{cases} , \tag{4.6}$$

que permite definir el elemento de combinación como

$$\Psi\left(\Theta\right) = \begin{cases} \theta_r & \text{si } \Psi_r = 1 \text{ y } \Psi_{i \neq r} = 0 \\ \theta_r & \text{si } \Psi_r = \Psi_s = 1 \text{ y } z^r > z^s \quad r \neq s \\ \emptyset & \text{si } \Psi_r = 0 \quad \forall r \end{cases} . \tag{4.7}$$

Asumiendo que las salidas de todos los nodos de dicotomía poseen un mismo rango de valores, sus respuestas son comparables. Esta metodología, denominada 'el ganador se queda con todo — del inglés, *winner-takes-all* — , evita cualquier ambigüedad si la etiqueta es asignada por la dicotomía que emite un valor de salida mayor, excepto en el caso de ambigüedad extrema con todas las predicciones de signo negativo. Pero, ¿permite esta variación asegurar una correcta clasificación final por parte de la máquina de aprendizaje multiclase? Si existe un empate en el cómputo de predicciones es seguro que alguno de los nodos de dicotomía está emitiendo una opinión equivocada. ¿Es adecuado asegurar que se equivoca aquel nodo que "grita menos", aquel cuyo valor numérico de salida es menor? En [Mayoraz and Alpaydin, 1999] se defiende que en caso de utilización de nodos SVMC no es de ninguna manera seguro trabajar con la comparación de salidas numéricas para obtener la respuesta final. La razón es que la escala de salida de cada una de las SVMC está determinada para conseguir que el hiperplano separador se halle en forma canónica, es decir que las salidas sobre los vectores soporte sean ± 1, una escala que no es robusta pues depende de sólo unos pocos puntos. En cambio, los esquemas de votación estándares son independientes del rango de salidas de cada clasificador binario.

Variaciones de los Esquemas de Votación sobre dicotomías 1-*v*-1 La combinación de 1-v-1 clasificadores binarios mediante un esquema de votación por unanimidad elimina la posibilidad de situaciones de empate, pero en numerosas ocasiones deja entradas sin etiquetar debido a la necesidad de que todas las dicotomías implicadas en la designación de la etiqueta realicen una predicción correcta. La menor exigencia de los otros dos esquemas de votación por mayoría para adjudicar una etiqueta es utilizada como recurso para asegurar el etiquetado, aunque por contra se crean situaciones de empate que de nuevo son tradicionalmente solucionadas mediante el uso del método *winner-takes-all*, poco práctico si la dicotomías implicadas son SVMCs.

4.2.2 Otras arquitecturas en paralelo

Los intentos de mejora en la respuesta se centran en variaciones sobre el método de reconstrucción, en interpretaciones probabilísticas de las salidas numéricas de los biclasificadores y en modificaciones en la agrupación de clases para realizar la dicotomía, denominada codificación.

Clasificación por Parejas

En el método de reconstrucción denominado "por parejas" — en inglés, *Pairwise Classification* —[Kressel, 1999] se interpretan las salidas de las dicotomías por medio de una especie de grado de pertenencia a la clase. Si se renombra la salida signo z^k correspondiente a cada uno de los L 1-v-1 clasificadores como z^{ij} para indicar que la k-ésima dicotomía separa las clases θ_i y θ_j, entonces es posible definir $\Psi_r = \sum_j z^{rj}$, con $z^{ji} = -z^{ij}$, como la suma de votos recibidos a favor y en contra por la clase θ_r por todos las dicotomías donde θ_r está implicada. Es decir, si la (r, s)-ésima dicotomía emite $z^{rs} = +1$, entonces suma una unidad a Ψ_r, mientras que resta una en Ψ_s. Se comprueba empíricamente que este método de reconstrucción produce mejores fronteras de decisión que la versión estándar con dicotomías 1-v-r, en el sentido de obtener espacios finales de ambigüedad más reducidos, aunque el estudio se restringe a fronteras lineales con problemas de sólo 3 clases.

Dicotomías ECOC

Tal como se recuerda en [Alpaydin and Mayoraz, 1998], los dos principales inconvenientes de crear un esquema de descomposición basado en dicotomías 1-v-1 son que el número de clasificadores es $\mathcal{O}(K)$ y que cada uno de ellos es entrenado con datos extraídos de sólo dos clases del conjunto de entrenamiento por lo que la varianza es mayor y no da información sobre el resto de clases. Sería oportuno un número menor de dicotomías que fueran entrenadas sobre todo el conjunto de entrenamiento.

Definición 4.7. *Dentro de una arquitectura de máquina de clasificación multiclase, se denomina* codificación estándar *a cada una de las posibles particiones de todo el conjunto de clases* $\mathcal{Y} = \{\theta_1, \ldots, \theta_{K>2}\}$ *en dicotomías que asignan etiquetas positivas* $t_p^k = 1$ *a los patrones de entrenamiento de un cierto subconjunto de clases,* $\mathcal{Y}_+$*, y etiquetas negativas* $t_p^k = -1$ *a los patrones de entrenamiento representantes del resto de clases* $\mathcal{Y}_- = \mathcal{Y} \backslash \mathcal{Y}_+$.

Siguiendo esta definición, un esquema de descomposición 1-v-r puede entenderse como una codificación con dicotomías del estilo $\mathcal{Y}_+ = \theta_i$, $\mathcal{Y}_- = \mathcal{Y}\backslash\theta_i$, $\forall i$.

Aplicando las ideas de la técnica *Error Correcting Output Codes* — en corto, ECOC — [Dietterich and Bakiri, 1995] que utiliza la codificación estándar para obtener robustez contra fallos en las dicotomías, las columnas en la matriz de descomposición $\mathbf{D}$ generada por la Ecuación 4.2, deberían ser tan diferentes como fuera posible en términos de la distancia Hamming para añadir redundancia. Por ejemplo, con $K = 4$ el número de dicotomías a construir, $2^{(K-1)} - 1$, correspondería con la matriz de descomposición

$$\mathbf{D}_{ECOC} = \begin{pmatrix} +1 & -1 & -1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & -1 & +1 & +1 \\ +1 & +1 & -1 & -1 \\ +1 & +1 & -1 & +1 \\ +1 & +1 & +1 & -1 \end{pmatrix} , \tag{4.8}$$

es decir, los conjuntos de entrenamiento modificados $\mathcal{T}^k$ para entrenar cada dicotomía se corresponden con el conjunto completo $\mathcal{T}^*$, mientras las dicotomías construidas son un subconjunto de todas las posibles combinaciones entre clases. Para conseguir reducir el número final de dicotomías se diseña un cierto algoritmo que establece su construcción de forma incremental hasta conseguir el grado de precisión que se desee sobre el conjunto de entrenamiento [Alpaydin and Mayoraz, 1998].

El principal inconveniente de este método está en la cota máxima de dicotomías que se pueden construir, $2^{(K-1)} - 1$. Si bien cada dicotomía posee la ventaja de ser entrenada sobre todo el espacio de entrenamiento, el incremento en el grado de precisión cuando se está cercano a valores límite comporta un aumento muy significativo en el número de dicotomías a construir. En el caso de un problema con $K = 10$ clases, mientras un esquema 1-v-1 necesita construir 45 clasificadores binarios, un esquema de dicotomías ECOC puede llegar a 511.

Se ha de hacer notar que la presente variación ECOC ha introducido una nueva propiedad de mejora que sería deseable que dispusieran las arquitecturas de clasificación multiclase, la robustez de la salida final frente a fallos en las predicciones. Aunque las estructuras habituales y sus variaciones basan su esfuerzo en la eliminación de empates entre votos, lo que conlleva implícitamente que alguna dicotomía

ha emitido un juicio equivocado, no existe ninguna variación tradicional que trabaje explícitamente con errores en las predicciones y delimite sus consecuencias sobre el resultado final.

Dicotomías ECOC Generalizadas

En la matriz de descomposición (4.8) en dicotomías ECOC original, $\mathbf{D}_{ECOC}$, no existe la posibilidad de no consideración de alguna clase por alguno de los clasificadores binarios, tal como puede hacerse con $\mathbf{D}_{1-v-1}$. En [Allwein et al., 2000] se generaliza el esquema ECOC permitiendo esta posibilidad de descomposición.

Definición 4.8. *Dentro de una arquitectura de máquina de clasificación multiclase, se denomina* codificación generalizada *a cada una de las posibles particiones de un subconjunto de las clases* $\mathcal{Y} = \{\theta_1, \ldots, \theta_{K>2}\}$, $\mathcal{Y}_{sub} \subseteq \mathcal{Y}$,*en dicotomías que asignan etiquetas positivas* $t_p^k = 1$ *a los patrones de entrenamiento de un cierto subconjunto de clases,* $\mathcal{Y}_+ \subsetneq \mathcal{Y}_{sub}$, *y etiquetas negativas* $t_p^k = -1$ *a los patrones de entrenamiento representantes del resto de clases en* $\mathcal{Y}_{sub}$, $\mathcal{Y}_- = \mathcal{Y}_{sub} \backslash \mathcal{Y}_+$.

Siguiendo esta definición, un esquema de descomposición 1-v-1, que no podía codificarse con la codificación estándar, ahora puede entenderse como una codificación con dicotomías del estilo $\mathcal{Y}_+ = \theta_i$, $\mathcal{Y}_- = \theta_j$, $\forall i < j$.

Como método de reconstrucción los autores ofrecen dos nuevas posibilidades: una generalización de la clasificación "por parejas" con sólo suma de votos positivos, o bien la misma generalización pero con la suma de los valores numéricos de salida de las dicotomías. De entre las conclusiones obtenidas en el estudio, que incluyen un análisis teórico del error de generalización multiclase, se ha de destacar que cuando se utilizaron SVMC como clasificadores binarios sobre una gran variedad de problemas del *UCI Repository* [Blake and Merz, 1998], la descomposición 1-v-r obtuvo a menudo resultados con niveles de error mucho más altos que cualquiera de los otras codificaciones testeadas, a pesar de ser la más comúnmente usada. Con respecto a la codificación 1-v-1, sus resultados fueron comparables a las codificaciones propuestas por los autores.

De nuevo la principal desventaja de este tipo de codificación de clases se encuentra en la gran cantidad de dicotomías que se deben realizar. Tal como admiten los autores, la codificación o elección de dicotomías con el fin de reducir su número se realizó maximizando un cierto valor ρ que minimiza la cota teórica de error de generalización, pero de forma aleatoria sobre un subconjunto amplio de todas las codificaciones posibles.

Salidas Numéricas Probabilísticas

Uno de los campos de investigación todavía abiertos en la teoría de aprendizaje estadístico es el intento de convertir la salida no calibrada de una SVMC en una salida probabilística de forma que la máquina compute una probabilidad *a posteriori*. En [Wahba, 1999] por ejemplo se propone usar la función logística

$$P\left(clase|entrada\right) = \frac{1}{1+\exp\left(-f\left(\mathbf{x}\right)\right)}\,, \tag{4.9}$$

donde $f\left(\mathbf{x}\right)$ es el hiperplano generado por la SVMC. Otras posibilidades y resultados pueden encontrarse en [Platt, 1999b], [Kwok, 1999].

Asumiendo que cada dicotomía sea capaz de computar una probabilidad, realizando para ello las modificaciones oportunas sobre la salida numérica, puede usarse de nuevo como notación de salida numérica z^{ij}, para indicar que la k-ésima dicotomía separa las clases θ_i y θ_j, representando ahora una probabilidad, $z^{ij} \in [0,1]$, con $z^{ij}=1$ para indicar la seguridad de que la entrada pertenece a la clase θ_i y $z^{ij}=0$ si es segura su pertenencia a la clase θ_j. Igual que en el caso de clasificación "por parejas" se puede establecer una probabilidad *a posteriori*, antes se definió como grado de pertenencia, para cada clase

$$p_i = \frac{2}{K\left(K-1\right)}\sum_{j\neq i} z_{ij}\,, \tag{4.10}$$

que permite adjudicar como etiqueta la de aquella clase que posea una probabilidad p_i mayor [Moreira and Mayoraz, 1998].

Una posibilidad de corregir las predicciones de los clasificadores binarios 1-v-1, cuya salida posee mucha varianza, para obtener mayor robustez consiste en añadir unos clasificadores extras. Por cada 1-v-1 clasificador binario separando las clases θ_i y θ_j, se construiría un clasificador adicional, f_{ij}, que separaría estas dos clases del resto de clases, $\mathcal{T}_+^{ij} = \mathcal{T}_i \cup \mathcal{T}_j$ y $\mathcal{T}_-^{ij} = \mathcal{T}^* \backslash \left(\mathcal{T}_i \cup \mathcal{T}_j\right)$, por lo que el número de dicotomías final sería $L = K\left(K-1\right)$, y los nuevos clasificadores poseerían una matriz de descomposición $\mathbf{D}$, por ejemplo para el caso $K=4$

$$\mathbf{D}_{extra} = \begin{pmatrix} +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ -1 & +1 & +1 & -1 \\ -1 & +1 & -1 & +1 \\ -1 & -1 & +1 & +1 \end{pmatrix} \quad . \tag{4.11}$$

Se trata pues de una versión semejante a la de las dicotomías ECOC pero no trabajando todos los clasificadores en paralelo sino formando dos capas. Si se define

q_{ij} como la salida probabilística de un clasificador adicional f_{ij}, entonces se puede modificar la probabilidad *a posteriori* para cada clase (4.10) como

$$p_i = \frac{2}{K\,(K-1)} \sum_{j \neq i} z_{ij} q_{ij}\,, \tag{4.12}$$

con la intención que los clasificadores correctores adicionales hagan perder significancia a los z_{ij} irrelevantes y permitan mejorar la calidad de la estimación de las probabilidades *a posteriori* p_i.

Debe resaltarse que la demanda de cálculo de dicotomías es elevada, el doble que en una descomposición 1-v-1 tradicional sobre todo el conjunto de entrenamiento, y el cálculo de la probabilidad final como producto de probabilidades, las de las dicotomías originales por las de las dicotomía extras, provoca la exportación, aunque atenuado, del posible error original hacia la salida final.

Normalización de las Salidas Numéricas

En [Mayoraz and Alpaydin, 1999] es propuesta una técnica geométrica de normalización para conseguir que las salidas de todas las SVMCs se hallen en el mismo rango de valores que consiste en definir un factor de escala π_ω^k con el fin de obtener $\|\omega\|_2 = 1$ para todos los hiperplanos solución $f_k\,(\mathbf{x}, \omega)$ generados por los 1- v-r clasificadores.

La propuesta de solución de estos autores no se tendrá en consideración por dos motivos: el elemento de normalización de las salidas se convierte en un elemento crítico para asegurar las prestaciones de la máquina de multiclasificación, y por otra parte los nodos de dicotomía continúan siendo del estilo 1-v-r lo que convierte el proceso de aprendizaje en altamente desigual, en caso de un número similar de patrones de aprendizaje para cada clase y un número de clases sólo moderadamente alto, por ejemplo $K > 5$.

4.2.3 Arquitecturas en árbol

Además de la posibilidad de trabajar con esquemas de descomposición en paralelo, pueden construirse arquitecturas de clasificadores multiclase combinando 1-v-1 SVMCs con árboles de decisión [Quinlan, 1986], que están diseñados para manejar problemas con muchas clases.

Arquitectura DAGSVM

En [Platt et al., 2000] es definido un especial tipo de grafo que permite su combinación con dicotomías,

Definición 4.9. *Se define el* Grafo Acíclico Dirigido de Decisión *—DDAG, del inglés Decision Directed Acyclic Graph — , como un grafo con raíz cuyas ramas, de salida binaria, no poseen orientación ni ciclos, de K hojas con etiquetas diferentes y con* $L = K\,(K-1)\,/2$ *nodos internos separando diferentes pares de clases.*

Los autores demuestran que si los nodos son separadores lineales entonces se puede controlar la amplitud o capacidad de un DDAG mediante la anchura del margen, por lo que aconsejan utilizarlo con nodos SVMC. También se asegura que en tal caso, el tiempo requerido para entrenar un DDAG con 1-v-1 SVMCs es aproximadamente sólo el doble a aquel utilizado para entrenar una sola máquina 1-v-r SVMC.

En los estudios empíricos realizados sobre el conjunto *USPS* de dígitos escritos manualmente y el conjunto *Letter* del *UCI Repository*, los niveles de error del nuevo algoritmo fueron similares al 1-v-r SVMC tradicional y al 1-v-1 SVMC con esquema de votación por mayoría simple, aunque mostró un tiempo de evaluación y entrenamiento inferior.

Sobre esta nueva arquitectura deben realizarse dos comentarios. Una razón que induce a descartar de pleno este tipo de arquitectura es su nula robustez ante errores en los nodos de dicotomía. Un error en uno de estos nodos implica el error sobre la clasificación final. Esta realidad es la que lleva a la segunda observación: se ha destacado anteriormente por numerosos autores la obtención de mejores resultados con la metodología 1-v-1 que con la 1-v-r, por lo que si el estudio empírico hubiera sido más exhaustivo sería de esperar, dada la nula robustez del algoritmo, que el comportamiento del método 1-v-1 fuera mejor que el de la arquitectura propuesta. Si bien es positivo conseguir un menor costo computacional, las respuestas de las dicotomías continúan teniendo excesiva varianza.

Combinando RLP con SVMC

En el trabajo con grandes conjuntos de datos, los árboles de decisión que implementan en sus nodos decisiones simples pueden ser enormes, mientras que una SVMC puede ser interpretada como un árbol de decisión muy simple que implementa una decisión que no puede ser interpretada, es una caja negra. Con la intención de llegar a un compromiso entre ambas visiones, en [Bennett, 1999] se examina la relación entre el problema de programación cuadrática —QP, del inglés *Quadratic Programming* — asociado a las SVMC y el uso del problema dual de la Programación Lineal Robusta [Bennett and Mangasarian, 1994] — RLP, del inglés *Robust Linear Programming* — que permite construir árboles de decisión simples con una excelente reducción de dimensionalidad, el Árbol de Decisión basado en Vectores Soporte — SVDT, del inglés *Support Vector Decision Tree* — cuyos nodos de decisión son SVMCs lineales.

La comparación de este método de clasificación multiclase con todos los anteriores no es posible pues su definición está demasiado cercana a los árboles de decisión.

De hecho la principal ventaja destacada por su autora es la posibilidad de extraer información a partir de la estructura del árbol, como se muestra en diversos problemas tipo sobre análisis de mercado.

4.3 Esquemas "Todas las Clases a la Vez"

Además de los métodos de {descomposición, reconstrucción}, otra posibilidad de trabajo con SVMCs para el caso de problemas de clasificación multiclase consiste en derivar una única máquina capaz de trabajar con todas las clases del conjunto de salida $\mathcal{Y}$ a la vez, que será denominada KSVMC. Máquinas que poseen en la actualidad estas características de multiclasificación son: una generalización natural de una SVMC que es biclase por definición, una segunda construcción basada en un resultado de convergencia uniforme o una combinación de SVM con RLP.

4.3.1 Generalización de una SVMC

La máquina presentada en este apartado es una extensión natural de las SVMC biclases al caso multiclase, cuya formulación en términos similares ha sido propuesta de forma independiente por [Weston and Watkins, 1998] y [Vapnik, 1998] entre otros. Siguiendo la notación en [Weston and Watkins, 1998], si se escriben las clases de salida sólo como el subíndice que las identifica,

$$\mathcal{Y} = \{\theta_1, \ldots, \theta_{K>2}\} \Rrightarrow \mathcal{Y} = \{1, \ldots, K\} \,, \tag{4.13}$$

se puede generalizar el problema de optimización sobre el funcional de riesgo (3.6) de la página 53 a este otro

$$\arg\min R_{KSVMC}\left(\omega, \xi\right) = \frac{1}{2} \sum_{m=1}^{K} \|\omega_m\|_{\mathcal{F}}^2 + C \sum_{i=1}^{\ell} \sum_{m \neq y_i} \xi_i^m \,, \tag{4.14}$$

sujeto a las restricciones

$$\langle \omega_{y_i}, \mathbf{x}_i \rangle_{\mathcal{F}} + b_{y_i} \geq \langle \omega_m, \mathbf{x}_i \rangle_{\mathcal{F}} + b_m + 2 - \xi_i^m \,, \tag{4.15}$$

$$\xi_i^m \geq 0 \quad i = 1, \ldots, \ell \quad m \in \mathcal{Y} \backslash y_i \,, \tag{4.16}$$

y teniendo como función de decisión una estructura del estilo

$$f\left(\mathbf{x}\right) = \arg\max_{K} \left[\langle \omega_m, \mathbf{x}_k \rangle_{\mathcal{F}} + b_k\right] \quad k = 1, \ldots, K \,. \tag{4.17}$$

Introduciendo multiplicadores de Lagrange y la notación

$$c_i^n = \begin{cases} 1 & \text{si } y_i = n \\ 0 & \text{si } y_i \neq n \end{cases} \,, \tag{4.18}$$

$$A_i = \sum_{m=1}^{K} \alpha_i^m \,, \tag{4.19}$$

si se realizan algunas substituciones, se obtiene el enunciado dual de Wolfe del problema de optimización original: Hallar multiplicadores $\alpha_i^m \geq 0$ que hagan mínimo el funcional

$$W(\alpha) = \frac{1}{2} \sum_{i,j=1}^{\ell} \sum_{m=1}^{K} \left[c_j^{y_i} A_i A_j + \alpha_i^m \alpha_j^m - 2\alpha_i^m \alpha_j^{y_i} \right] \cdot k(\mathbf{x}_i, \mathbf{x}_j) - 2 \sum_{i=1}^{\ell} A_i \,, \tag{4.20}$$

sujeto a

$$0 \leq \alpha_i^m \leq C, \quad \alpha_i^{y_i} = 0 \quad i = 1, \ldots, \ell \quad m \in \mathcal{Y} \backslash y_i \,, \tag{4.21}$$

$$\sum_{i=1}^{\ell} \alpha_i^n = \sum_{i=1}^{\ell} c_i^n A_i \quad n = 1, \ldots, K \,, \tag{4.22}$$

La función de decisión escrita en función de los hiperplanos separadores solución puede ser expresada como

$$f(\mathbf{x}) = \arg\max_n \left[\left(\sum_{i:y_i=n} A_i - \sum_{i:y_i \neq n} \alpha_i^n \right) \cdot k(\mathbf{x}_i, \mathbf{x}) + b_n \right] . \tag{4.23}$$

El principal inconveniente de esta formulación está en que el problema de optimización que se requiere resolver es cuadrático con $\ell(K-1)$ variables sujetas a $2\ell(K-1) + 2K$ restricciones de desigualdad. Así, aunque no se afirma de forma justificada, sí se apunta que los experimentos empíricos muestran que este tipo de solución es más lenta que aquellas basadas en una descomposición 1-v-r. Por otra parte, no existe ninguna inclusión de técnicas que mejoren la robustez del sistema ni ningún estudio teórico sobre la cota de error.

Con la finalidad de reducir el tiempo computacional también se deriva en el trabajo de [Weston and Watkins, 1998] un clasificador tipo *K*SVMC de programación lineal que aunque resulta en una máquina con un número menor de vectores soporte, obtiene niveles de error mayores que las máquinas de multiclasificación tradicionales.

4.3.2 *K*SVMC sobre Resultado de Convergencia Uniforme

Utilizando el reconocido resultado de [Bartlett, 1998], en [Guermeur et al., 1999] se obtiene una extensión de la cota de generalización del funcional de riesgo estructural (1.15) de la página 30 para el caso de modelos discriminantes multiclase que permite a los autores introducir en [Guermeur et al., 2000] una nueva estructura *K*SVMC basada en la ley fuerte uniforme de los grandes números.

Usando la notación de subíndices para las clases del conjunto de salida como en la Subsección anterior, la generalización del hiperplano separador aumenta según el resultado de convergencia uniforme si se resuelve el problema de optimización cuadrática restringido modificando el funcional de riesgo (3.6) por este otro

$$\arg\min R_{Unif}\left(\omega,\xi\right)=\frac{1}{2}\sum_{j=1}^{K-1}\sum_{k=j+1}^{K}\left\|\omega_j-\omega_k\right\|_{\mathcal{F}}^2+C\sum_{i=1}^{\ell}\sum_{m\neq y_i}\xi_i^j\,, \tag{4.24}$$

sujeto a las restricciones

$$\left\langle\omega_{y_i}-\omega_j,\mathbf{x}_i\right\rangle_{\mathcal{F}}+b_{y_i}-b_j\geq 1-\xi_i^j\,, \tag{4.25}$$

$$\xi_i^j\geq 0\quad i=1,\ldots,\ell\quad j\in\mathcal{Y}\backslash y_i\,, \tag{4.26}$$

tomando como función de decisión una estructura del estilo

$$f\left(\mathbf{x}\right)=\arg\max_k\left[\left\langle\omega_j,\mathbf{x}_k\right\rangle_{\mathcal{F}}+b_k\right]\quad k=1,\ldots,K\,. \tag{4.27}$$

Introduciendo multiplicadores de Lagrange, si se realizan algunas substituciones, se obtiene el enunciado dual de Wolfe del problema de optimización original, que siguiendo la notación de (4.19) puede escribirse como: Hallar multiplicadores $\alpha_i^m\geq 0$ que hagan mínimo el funcional

$$W\left(\alpha\right)=\frac{1}{2K}\sum_{i,j=1}^{\ell}\sum_{m=1}^{K}\left[c_j^{y_i}A_iA_j+\alpha_i^m\alpha_j^m-2\alpha_i^m\alpha_j^{y_i}\right]\cdot k\left(\mathbf{x}_i,\mathbf{x}_j\right)-\sum_{i=1}^{\ell}A_i\,, \tag{4.28}$$

sujeto a

$$0\leq\alpha_i^m\leq C,\quad\alpha_i^{y_i}=0\quad i=1,\ldots,\ell\quad m\in\mathcal{Y}\backslash y_i\,, \tag{4.29}$$

$$\sum_{i=1}^{\ell}\alpha_i^n=\sum_{i=1}^{\ell}c_i^nA_i\quad n=1,\ldots,K\,. \tag{4.30}$$

En consecuencia, los hiperplanos separadores solución separando las clases m y n que permiten definir la función de decisión tienen por expresión

$$f\left(\mathbf{x}\right)=\frac{1}{K}\sum_{i=1}^{\ell}\left[\left(c_i^m-c_i^n\right)A_i-\left(\alpha_i^m-\alpha_i^n\right)\right]\cdot k\left(\mathbf{x}_i,\mathbf{x}_j\right)+b_m-b_n\,. \tag{4.31}$$

La gran diferencia en la construcción de la máquina *K*SVMC respecto el caso anterior reside en la forma que adopta la función decisión. Mientras en (4.23) la decisión se toma sobre los K hiperplanos separadores de una clase frente al resto, 1-v-r, en el presente desarrollo la decisión depende de los K^2 hiperplanos (4.31) que separan clases al estilo 1-v-1, por lo que esta perspectiva ofrece dos ventajas sobre la anterior: se elabora sobre una teoría que permite establecer una cota de error de generalización, y la máquina crea separaciones 1-v-1 habiendo trabajado sobre todo el conjunto de entrenamiento.

Observando los enunciados de optimización duales del caso precedente (4.20) y del actual (4.28), se observa que la sola diferencia es el factor $\frac{1}{K}$ que acompaña al primero de los términos, mientras que las restricciones son equivalentes en ambos enunciados. Un ejemplo que permite ilustrar su relación es el siguiente:

Ejemplo 4.10. *Si se toma por núcleo en (4.20) una función tan habitual como la gaussiana*

$$k\left(\mathbf{x}_i, \mathbf{x}\right) = \exp\left(-\frac{\left\|\mathbf{x} - \mathbf{x}_i\right\|^2}{\sigma^2}\right), \tag{4.32}$$

de amplitud σ, una forma de conseguir la equivalencia entre (4.20) y (4.28) sería posible si se consiguiera definir la amplitud $\widetilde{\sigma}$ en el núcleo de (4.28) como

$$\widetilde{\sigma}^2 = \sigma^2 \cdot \frac{\left\|\mathbf{x} - \mathbf{x}_i\right\|^2}{\left\|\mathbf{x} - \mathbf{x}_i\right\|^2 - \sigma^2 \ln K} > \sigma^2 . \tag{4.33}$$

Esto permite deducir que en el segundo caso se obtendrían soluciones con funciones gaussianas más amplias que en el primero, lo que lleva a soluciones más generalizadoras.

De nuevo, el principal inconveniente de esta formulación está en la magnitud del problema de optimización que se requiere resolver. Se trata de un problema de programación cuadrática con $\ell\left(K-1\right)$ variables sujetas a $2\ell\left(K-1\right)+2K$ restricciones de desigualdad y en este caso el tiempo de evaluación es mayor pues se han de comparar K^2 valores emitidos por los hiperplanos de separación. Su robustez debe ser comparable a la de un esquema de descomposición 1-v-1 tradicional.

4.3.3 Combinando RLP con SVMC

En [Bredensteiner and Bennett, 1999] se muestra una versión híbrida entre un método de programación lineal, el RLP, en su versión multiclase, k-RLP, y el método SVM basado en programación cuadrática con el fin de obtener un nuevo algoritmo de clasificación multiclase, M-SVM, capaz de construir funciones de clasificación lineales a trozos.

Utilizando la misma notación de subíndices que en los apartados anteriores para las clases del conjunto de salida, el problema de optimización cuadrática restringido modificado a resolver puede expresarse como

$$\begin{aligned}\arg\min R_{M-SVM}\left(\omega, \xi\right) =& \frac{1}{2}\left[\sum_{m=1}^{K}\left\|\omega_m\right\|_{\mathcal{F}}^2 + \sum_{m=1}^{K-1}\sum_{n=m+1}^{K}\left\|\omega_m - \omega_n\right\|_{\mathcal{F}}^2\right] + \\ &+ C\sum_{i=1}^{\ell}\sum_{m\neq y_i}\xi_i^m\end{aligned} \tag{4.34}$$

sujeto a las restricciones

$$\langle \omega_{y_i} - \omega_j, \mathbf{x}_i \rangle_{\mathcal{F}} + b_{y_i} - b_j \geq 1 - \xi_i^j \,, \tag{4.35}$$

$$\xi_i^j \geq 0 \quad i = 1, \ldots, \ell \quad j \in \mathcal{Y} \backslash y_i \,, \tag{4.36}$$

obteniéndose finalmente como función de decisión

$$f(\mathbf{x}) = \arg\max_n \left[\left(\sum_{i:y_i=n} A_i - \sum_{i:y_i \neq n} \alpha_i^n \right) \cdot k(\mathbf{x}_i, \mathbf{x}) + b_n \right] . \tag{4.37}$$

Como puede observarse, el nuevo método tiene en el funcional de riesgo definido por la Ecuación 4.34 una conjunción de los métodos anteriores, 4.14 y 4.24, mientras que el resultado adopta un estilo 1-v-r. Por tanto, aunque posee en su funcional de riesgo el término $\|\omega_m - \omega_n\|_{\mathcal{F}}^2$ que favorece la generalización, sus resultados son muy semejantes a la natural extensión del primer KSVMC considerado, adoleciendo de los mismos problemas de excesiva dimensión del problema tratado.

4.4 Estudio de la Robustez

Para estudiar la robustez de una arquitectura multiclase se tendrá en consideración sólo métodos de votación como combinadores de predicciones signatorias. La razón principal está en la poca credibilidad que poseen las predicciones numéricas cuando los nodos de dicotomía son SVMCs, como ya ha sido apuntado y justificado con anterioridad.

Definición 4.11. *Sea* $\mathbf{x} \in \mathcal{X}$ *una entrada cuya salida* θ_m *es conocida. Sea* $\varepsilon_{rob}(\mathbf{x}, F) = \#f_m^{err}/L_m$ *la proporción que representa el número de dicotomías implicadas con la clase* θ_m *que emiten una salida equivocada respecto al número total de nodos implicados, aún siendo la salida final de la máquina correcta,* $F(\mathbf{x}) = \theta_m$*. Si se define el* factor de robustez *como*

$$\varepsilon_{rob}(F) = \arg\min_{\mathbf{x}} \varepsilon_{rob}(\mathbf{x}, F) \quad \forall \mathbf{x} \in \mathcal{X} \,, \tag{4.38}$$

se dirá que una arquitectura de {*descomposición, reconstrucción*} *para clasificación multiclase* $\mathbf{A}_1$ *es más* robusta *que otra* $\mathbf{A}_2$ *si*

$$\varepsilon_{rob}^1 = \min_{F \in \mathbf{A}_1} \varepsilon_{rob}^1(F) > \min_{F \in \mathbf{A}_2} \varepsilon_{rob}^2(F) = \varepsilon_{rob}^2 \,, \tag{4.39}$$

donde los superíndices indican la arquitectura a la que hace referencia la proporción.

Básicamente el factor de robustez trata de indicar, para el caso peor, cuántas dicotomías implicadas con la clase a la que pertenece la entrada pueden llegar a equivocarse dentro de un esquema de arquitectura multiclase sin que la salida final se vea alterada.

Proposición 4.12. *Una arquitectura de multiclasificación {1-v-r SVMC, votación} posee un factor de robustez nulo,* $\varepsilon_{rob} = 0$.

Demostración. Según la función intérprete (4.4) de los métodos de votación para descomposiciones 1-v-r el número máximo de votos que puede recibir una clase es 1. Sea θ_m la salida correcta. Si la dicotomía m-v-r se equivoca, todas las clases se quedan sin votos y se crea ambigüedad, $F(\mathbf{x}) = \emptyset \neq \theta_m$. Si es cualquier otro el nodo equivocado, entonces hay más de una clase con 1 voto y la ambigüedad de nuevo crea salida nula. □

Proposición 4.13. *Una arquitectura de multiclasificación {1-v-1 SVMC, votación} posee un factor de robustez nulo,* $\varepsilon_{rob} = 0$.

Demostración. En el peor de los casos, si θ_m es la salida correcta, una máquina sin fallos puede tener como votos asignados a esta clase $\Psi_m = L - 1$ y existir otra clase $\theta_{n \neq m}$ con $\Psi_n = L - 2$ votos. Basta con que el (n, m)-ésimo 1-v-1 nodo de dicotomía se equivoque para que el recuento sea a la inversa. □

Aunque la demostración se basa en un nodo muy particular, lo cierto es que no hace falta rebuscar para conseguir una salida final equivocada. Por ejemplo, si en una arquitectura multiclase {1-v-1 SVMC, votación} con $K = 5$, se comete una equivocación sobre uno de los nodos que trabajan sobre la clase final adecuada, de las 64 combinaciones de resultados finales posibles, sólo 14 (21.9%) continúan siendo el adecuado, 8 (12.5%) equivocan la clase y 42 (65.6%) emitirían una respuesta nula en la votación por mayoría absoluta debido a un empate.

La mejora que muestra de forma empírica la clasificación "por parejas" consiste en una reducción del espacio de ambigüedad en la clasificación. Sin embargo, tal como se ha definido el factor de robustez, que considera como máquina inicial una cuyas dicotomías no cometen errores sobre la entrada $\mathbf{x}$ en consideración — no basta con no cometer error sobre el conjunto de aprendizaje —, resulta cierto el siguiente corolario, cuya demostración es obvia.

Corolario 4.14. *Una arquitectura de multiclasificación {1-v-1 SVMC, votación "por parejas"} posee un factor de robustez nulo,* $\varepsilon_{rob} = 0$.

El cálculo del factor de robustez sobre métodos de descomposición basados en ECOC no es posible de forma general debido a que el conjunto de dicotomías realizadas se elabora de forma aleatoria o siguiendo un cierto algoritmo empírico de minimización de errores de clasificación. Ante esta imposibilidad, y tal como se comprobará más adelante para casos particulares, se dará por buena la suposición de que la habilidad de la metodología ECOC para generar clasificaciones robustas se mantiene aún siendo sólo un subconjunto de dicotomías quien actúa en la arquitectura de multiclasificación.

Respecto a las arquitecturas con reconstrucción en árbol se puede afirmar,

Proposición 4.15. *Una arquitectura DAGSVM, que bien podría notarse como {1-v-1 SVMC, DDAG}, posee un factor de robustez nulo,* $\varepsilon_{rob} = 0$.

Demostración. Debido a la estructura en árbol, al menos una de las dicotomías consultadas durante la fase de evaluación está implicada en la clase a la que pertenece la entrada $\mathbf{x}$ en consideración. Bastaría con suponer que esta dicotomía está equivocada para obtener una respuesta final equivocada. □

De nuevo podría pensarse por la demostración que el caso peor no es el caso habitual. Puede afirmarse que la posibilidad de emitir una respuesta final equivocada aumenta cuanto más arriba en la estructura del árbol se produce la equivocación y cuanto menos nodos implicados en la clase correcta se encuentran en los nodos inferiores del árbol. En un caso extremo, si se considera una sola equivocación y esta se produce en la raíz de árbol, la respuesta final seguro que también será equivocada. Por otra parte, si el nodo equivocado está en la rama final, entonces la mayoría de veces la respuesta final será correcta.

Haciendo uso de esta propiedad, con el fin de obtener mayor robustez en las salidas, una posible mejora del método podría conseguirse construyendo un árbol para cada clase de forma que en cada árbol F_i la clase θ_i sea evaluada en los nodos más bajos posibles de la estructura arbórea. Aunque el tiempo de evaluación aumenta, este coste no es significativo pues todos los nodos ya han sido entrenados y sólo es necesario permitir diferentes reordenaciones.

En el caso de arquitecturas multiclase considerando todas las clases a la vez, el estudio de robustez no es posible sobre la formulación original pues la función de decisión actúa sobre valores numéricos. Si una variación es introducida de forma que los valores numéricos sean transformados en votos, entonces la estructura que adoptan los diferentes algoritmos es del tipo {1-v-1, votación} o {1-v-r, votación}, que poseen factor de robustez nulo.

4.5 En Resumen

El contenido del presente Capítulo corresponde a un estudio detallado de las diferentes arquitecturas de clasificación multiclase creadas sobre una base de clasificadores diseñados para problemas de clasificación binaria, como es el caso de las SVMC.

Las arquitecturas multiclase de {descomposición, reconstrucción} con esquema de descomposición en paralelo son las más utilizadas cuando se trabaja con nodos de dicotomía tipo SVMC. En el método de descomposición, es posible codificar la

información en elementos binarios $\{-1, +1\}$ mediante tres esquemas principalmente: los más tradicionales 1-v-r y 1-v-1, y el ECOC. Este último puede usarse en el modo estándar, generalizado o en doble capa si las salidas de las dicotomías se interpretan como probabilidades.

Los métodos de reconstrucción basados en votación por signo tienen muchas variantes, votos sólo positivos, votos positivos y negativos, cómputo global de todos los votos, cómputo de sólo los votos de los clasificadores implicados, ... La finalidad es establecer un grado de pertenencia de un valor de entrada a cada una de las posibles clases para acabar seleccionando el mayor.

También es posible utilizar en la reconstrucción, ya sea como ayuda para deshacer empates o desde el inicio, el valor numérico de las salidas parciales. Ya se comentó que no es una estrategia adecuada si se utilizan SVMCs, aunque los intentos de interpretar las salidas como probabilidades *a posteriori* o de normalización de las salidas intentan superar este inconveniente intrínseco en la definición de las SVMs.

Se ha de destacar que los estudios empíricos muestran la mayor efectividad del método 1-v-1 sobre el 1-v-r, aunque de este último deben ser superados los inconvenientes de: (a) gran número de nodos de dicotomía, (b) aprendizaje sólo sobre un subconjunto del total y (c) robustez contra fallos en las respuestas parciales. El método ECOC está siendo la variante más utilizada para intentar superar las desventajas (b) y (c) aunque ello supone un aumento muy significativo en el número de nodos de dicotomía y por tanto en tiempo de cálculo computacional.

Otras estructuras posibles son aquellas que disponen una reconstrucción en árbol, ya sea basadas en resultados teóricos de generalización que permiten reducir el tiempo de evaluación, o definidas como una combinación con métodos de programación lineal con objeto de poder extraer información.

Los clasificadores únicos considerando todas las clases a la vez también son una alternativa, aunque conllevan la resolución de un problema de optimización de gran dimensión. Tres alternativas han sido mostradas, cada arquitectura obtenida desde perspectivas diferentes, con problemas de optimización primal parecidos.

Por último, se ha definido un factor de robustez que permite analizar la posibilidad de incorporar fallos en los nodos de dicotomía sin que el resultado correcto final se vea afectado. Aunque la definición no es aplicable en todas las arquitecturas consideradas, se ha observado la nula robustez de la mayoría de máquinas, mientras que sobre aquellas consideradas más robustas, las de estructura ECOC, no es posible calcular el factor de robustez en forma general debido a la aleatoriedad en la elección de los nodos de dicotomía.

En la Tabla 4.1 se muestra una comparativa entre los diferentes métodos sobre la complejidad de las estructuras que son tratadas. Para ilustrar la comparación se ha

	Nodos de Dicotomía	Variables	Restricciones $\leq$
1-v-r	$K = 10$	$\ell = 1000$	$2\,(V+1) = 2002$
1-v-1	$\binom{K}{2} = 45$	$2\ell/K = 200$	$2\,(V+1) = 402$
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	$K\,(K-1) = 90$	$2\ell/K, \ell$	$2\,(V+1)$
DAGSVM	$\binom{K}{2} = 45$	$2\ell/K = 200$	$2\,(V+1) = 402$
*K*SVMC 1	1	$\ell\,(K-1) = 9000$	$2\,(V+K) = 18020$
*K*SVMC 2	1	$\ell\,(K-1) = 9000$	$2\,(V+K) = 18020$
M-SVM	1	$\ell\,(K-1) = 9000$	$2\,(V+K) = 18020$
Deseable	$\leq \binom{K}{2} = 45$	$\ell = 1000$	$2\,(V+1) = 2002$

Tabla 4.1: Comparativa sobre complejidad de arquitecturas multiclase.

	Robustez	Tiempo CPU	Cota teórica	$\mathcal{T}^k$
1-v-r	nula	elevado	NO	$\mathcal{T}^*$
1-v-1	nula	medio	NO	$\mathcal{T}_i \cup \mathcal{T}_j$
ECOC est.	alta?	elevado	NO	$\mathcal{T}^*$
ECOC gener.	alta?	medio/elevado	NO	$\mathcal{T}_{sub} \subseteq \mathcal{T}^*$
ECOC capas	media?	elevado	NO	$\mathcal{T}_i \cup \mathcal{T}_j, \mathcal{T}^*$
DAGSVM	nula	bajo	SI	$\mathcal{T}_i \cup \mathcal{T}_j$
*K*SVMC 1	nula	muy elevado	NO	$\mathcal{T}$
*K*SVMC 2	nula	muy elevado	SI	$\mathcal{T}$
M-SVM	nula	muy elevado	NO	$\mathcal{T}$
Deseable	alta	bajo/medio	SI	$\mathcal{T}^*$

Tabla 4.2: Comparativa sobre características de arquitecturas multiclase.

tomado como ejemplo el caso de una clasificación multiclase con $\ell = 1000$ patrones de entrenamiento, divididos en $K = 10$ clases, teniendo cada clase la misma cantidad de representantes, $\ell_i = 100, \forall i$. También se indican las características viables que debería poseer una máquina de clasificación ideal.

En la Tabla 4.2 la comparativa se realiza sobre características más teóricas, sobre las cuales sería deseable que se mostrara un buen comportamiento. En el caso de uso de teoría ECOC, la robustez media-alta de las máquinas de aprendizaje se da por supuesta si se realiza una elección acertada de los nodos de dicotomía.