Universidad de Valladolid

Facultad de Ciencias

# Trabajo Fin de Grado

Grado en Matemáticas

## Fundamentos y Aplicaciones de la Teoría de Aprendizaje Estadístico

***Autor: Miguel Tereso del Río Almajano***

***Tutor/es: Eustasio del Barrio Tellado***

# Fundamentos y Aplicaciones de la Teoría de Aprendizaje Estadístico

Miguel Tereso del Río Almajano

18 de octubre de 2019

# Índice general

# Capítulo 1

# Introducción

Hoy en día hay una gran cantidad de datos que se deben analizar y el análisis humano resulta lento y costoso, a veces incluso inviable, por ello se tiene la necesidad de confiar en el análisis automático de los datos.

Imagínese que una empresa de seguridad instala cámaras que toman imágenes en blanco y negro cada cinco segundos las 24 horas del día. A esta empresa le interesa detectar cuándo en dichas imágenes aparece una persona, sin embargo saben que un experto tarda entre dos y tres segundos en distinguir si en la imagen aparece una persona o no. No es difícil comprobar que deben tener un experto contratado las 24 horas del día para cada dos cámaras de seguridad, lo cual es inasumible.

Podría pensarse que es una buena idea que un grupo de expertos analice todas las imágenes en blanco y negro posibles y ya nunca más se necesitaría la ayuda de expertos, pero es peor el remedio que la enfermedad. Tomando un tamaño estándar hay $(800 \times 600)^{256}$ imágenes posibles, es decir, si toda la humanidad trabajase sin descanso en este problema durante tres generaciones solo se clasificaría una de cada $2^{64}$ imágenes posibles.

Con vistas a automatizar el proceso de etiquetado, se pide la ayuda de expertos para etiquetar unas cuantas imágenes que se espera que sean suficientemente representativas.

Estas imágenes etiquetadas se introducen en un algoritmo con la esperanza de que este aprenda de forma automática una regla que etiquete suficientemente bien las imágenes.

Durante el desarrollo de esta memoria se garantizará que bajo ciertas condiciones se puede diseñar un algoritmo que devuelva una regla suficientemente buena.

El marco teórico en el que se trabaja es el iniciado por Vapnik en los años 70 (ver [6]). El trabajo de Vapnik, con la formalización final debida a Valiant (ver [5]) llegó poco después de que la aparición del algoritmo Perceptron (un procedimiento automático para encontrar un hiperplano separante entre dos conjuntos

de puntos) estimulase el interés tanto en las aplicaciones como en la teoría de la Inteligencia Artificial. La teoría de Vapnik surge como respuesta a la cuestión ¿qué significa que una máquina aprenda? o ¿qué cosas pueden ser aprendidas? La solución dada por Vapnik fue proponer un nuevo paradigma, diferente del de la estadística matemática clásica, aunque fuertemente relacionado con ella.

En este trabajo se propone el análisis de los fundamentos de esta teoría de aprendizaje estadístico (machine learning). Se comienza describiendo los elementos principales de esta teoría en el resto de la introducción. Aunque en el aprendizaje estadístico se consideran otros posibles problemas la teoría se presenta en el marco más concreto de la clasificación binaria (aprendizaje supervisado). Se discute la necesidad de limitar la tarea de aprendizaje a la consideración de modelos (clases de reglas) limitados y se completa la introducción presentando el paradigma de aprendizaje probablemente aproximadamente correcto (PAC).

El segundo capítulo está dedicado a presentar un resultado fundamental en esta teoría: la caracterización de las clases que son aprendibles PAC en términos de la dimensión de Vapnik-Chervonenkis. Este es un concepto combinatorio que mide la complejidad de la clase de reglas y está presente también en la aproximación más clásica a la estadística matemática (ver [2]).

El concepto de aprendizaje PAC resulta un tanto rígido y excluye a muchas clases de reglas interesantes. Por esta razón se han planteado versiones relajadas del concepto de aprendizaje PAC. Estas son estudiadas en el tercer capítulo de la memoria junto con el método de minimización del riesgo estructural.

La memoria se completa con el análisis de dos algoritmos que tienen un papel especialmente relevante en la teoría y la práctica del aprendizaje estadístico: las máquinas de soporte vectorial y las redes neuronales. Se presentarán resultados que proporcionan garantías probabilísticas sobre el correcto funcionamiento de los métodos.

La teoría y los algoritmos presentados en esta memoria están inspirados en el paradigma PAC y sus variantes. Este planteamiento es distinto de otros procedentes de la estadística matemática clásica que han generado otros métodos de gran importancia en el aprendizaje estadístico actual, tales como el Lasso o las redes elásticas (ver [1]). Sería muy interesante comparar la significión de las distintas garantías teóricas y el funcionamiento en la práctica de ambas aproximaciones, pero esto es una ardua tarea que queda fuera de los objetivos de esta memoria.

## 1.1. Descripción del problema

Se introducen en esta sección los conceptos y la notación que se utilizará para describir el problema desde un punto de vista matemático:

**Definición 1.1.1** *El conjunto de todos los atributos a etiquetar, en el ejemplo previo el conjunto de todas las imágenes posibles en blanco y negro, se conocerá*

*como el espacio de atributos y se denotará por $\mathcal{X}$.*

**Definición 1.1.2** *Al conjunto de todas las etiquetas que se pueden asignar a los elementos del espacio de atributos, en el ejemplo previo $\{-1, 1\}$ donde ($1 \equiv$ Aparece una persona) y ($-1 \equiv$ No aparece una persona), se le llamará espacio de etiquetas y se denotará por $\mathcal{Y}$.*

El problema descrito en la introducción es un problema de clasificación binaria dentro del marco del aprendizaje supervisado. Se pueden tratar muchos otros problemas de aprendizaje automático, pero durante el desarrollo de la memoria solo se tratarán problemas de aprendizaje supervisado. Además nos centraremos principalmente en problemas de clasificación binaria, es decir, problemas en los que el espacio de etiquetas tiene tamaño dos.

**Definición 1.1.3** *Se llama regla a cualquier función del espacio de atributos en el espacio de etiquetas. En el ejemplo previo cualquier función que asigna la etiqueta -1 ó 1 a cada una de las imágenes posibles. Normalmente las reglas se denotarán por $h$.*

Por cuestiones que se expondrán próximamente un algoritmo no escoge una regla del conjunto de todas las reglas posibles, sino de un subconjunto del mismo, que se llamará clase de reglas y se denotará normalmente por $\mathcal{H}$.

Se supone la existencia de una distribución subyacente que rige la aparición de los atributos etiquetados. Esta se denota normalmete por $D$. Evidentemente esta distribución es desconocida, pero se trata de hacer inferencias sobre ella. En nuestro caso por ejemplo las imágenes con valores muy distintos en pixeles consecutivos (imaginese un tablero de ajedrez enorme) son muy improbables, pero es imposible determinar la distribución exacta con la que aparecen las imágenes y con qué probabilidad en estas imágenes aparece una persona o no.

Para recabar información acerca de la distribución subyacente se llevan a cabo experimentos con el fin de conseguir una muestra, que se denota normalmente por $S$. En el ejemplo de la introducción la muestra son las imágenes etiquetadas por expertos. Se supone que $S = \{(x_1, y_1), ..., (x_m, y_m)\} \sim D$, es decir, que la muestra ha sido extraída de forma independiente e igualmente distribuidas (i.i.d.) de acuerdo con una distribución subyacente $D$. Se especifica a veces el tamaño de la muestra mediante la notación $S \sim D^m$.

Dada una clase de reglas $\mathcal{H}$ en esta memoria se llama algoritmo y se denota normalmente por $A$ a cualquier función que asigne una regla $h \in \mathcal{H}$ a cada muestra $S$. Un algoritmo simple y poco útil para el ejemplo que se está tratando sería aquel que dada una muestra que contuviese más imágenes con la etiqueta $-1$ que con la etiqueta 1 eligiese la regla que asigna a todas las imágenes la etiqueta $-1$ y dada cualquier otra muestra elija la regla que asigna a todas las imágenes la etiqueta 1.

Cabe destacar que algunos algoritmos tratados en este trabajo no siempre pueden ser llevados a cabo en la práctica, por el gran coste computacional que suponen. Durante esta memoria solo se estudian temas relacionados con la complejidad estadística y no con la complejidad computacional, por ello se recomienda al lector interesado en la complejidad computacional la lectura del Capítulo 8 de [3].

Dado un elemento del espacio de atributos $x$, para evaluar la precisión de una etiqueta escogida por una regla $h$ con respecto a la etiqueta real $y$, se debe escoger una función de pérdida que dependerá del problema a resolver. La función de pérdida se denota en general por $l(h,(x,y))$ y toma valores en los números reales.

Para el problema que se trata principalmente en esta memoria, la clasificación binaria, se utiliza principalmente la función de pérdida 0-1 ($l_{0-1}$):

$$l_{0-1}(h,(x,y)) = \mathbb{I}_{h(x)=y} = \begin{cases} 1 & si \quad h(x) = y \\ \\ 0 & si \quad h(x) \neq y \end{cases} \tag{1.1}$$

Para estudiar otros problemas se utilizan infinidad de funciones de pérdida. Entre las más conocidas se encuentra la función de pérdida cuadrática, $l_2$. Esta es muy utilizada cuando el espacio de etiquetas es $\mathbb{R}^d$, como por ejemplo en problemas de regresión lineal:

$$l_2(h,(x,y)) = (h(x) - y)^2. \tag{1.2}$$

Durante el desarrollo de la memoria también serán de utilidad las siguientes funciones de pérdida:

*Función de hinge,* $l_{hinge}$, definida por:

$$l_{hinge}(h,(x,y)) = \text{máx}\{0, \phi(h,(x,y))\}, \tag{1.3}$$

donde $F$ es una función con llegada en los reales.

*Función logística,* $l_{log}$, utilizada por sus buenas propiedades para el desarrollo de clasificadores lineales computables.

Cabe destacar que aunque no se enuncien, muchos de los resultados que se alcanzan durante el trabajo tienen su análogo para otras funciones de pérdida.

Para evaluar la precisión global de una regla sobre el espacio de atributos etiquetados, $\mathcal{X} \times \mathcal{Y}$, se utiliza el siguiente concepto:

**Definición 1.1.4** *(Riesgo) Siendo $D$ la distribución subyacente, el riesgo de una regla $h$ se denota por $L_D(h)$ y se define como:*

$$L_D(h) = \mathbb{E}_{(x,y)\sim D}(l(h,(x,y))).$$

Descrito el problema se procede a describir el objetivo: se busca un algoritmo que genere, basándose en la muestra, una regla que tenga el menor riesgo posible.

Se debe observar que ni conociendo la distribución subyacente está garantizado que se pueda etiquetar correctamente a todos los atributos. Prueba de ello es el siguiente ejemplo:

**Ejemplo 1.1.1** *Sean $X$ e $Y$ variables aleatorias de las que se sabe:*

· $\mathbb{P}[Y=0]=\mathbb{P}[Y=1]=\frac{1}{2}$

· $[X|Y=0]$ *sigue una distribución normal con media* 0 *y varianza* 1.

· $[X|Y=1]$ *sigue una distribución normal con media* $\mu$ *y varianza* 1.

*Se denotan las funciones de densidad de la $N(0,1)$ y de la $N(\mu,1)$ por $f_0$ y por $f_1$ respectivamente.*

*Dado el valor de $X$ se trata de inferir el valor de $Y$, equivalentemente, dado un elemento del espacio de atributos, $X$, se decide sobre su etiqueta (Y=0 ó Y=1). Se prueba a continuación que en este caso el valor de $X$ no aporta suficiente información como para determinar el valor de $Y$:*

*Sea $h$ una regla de clasificación binaria se define $H=\{x : h(x)=1\}$ y se calcula:*

$$\begin{aligned}\mathbb{P}[Y\neq h(X)] &= \mathbb{P}[Y=0]\mathbb{P}[h(X)=1]+\mathbb{P}[Y=1]\mathbb{P}[h(X)=0]\\ &= \frac{1}{2}\left[\int_{\mathcal{H}} f_0(x)dx+\int_{\mathcal{H}^C} f_1(x)dx\right]\\ &= \frac{1}{2}\int_{\mathbb{R}} f_0(x)\mathbb{I}_{\mathcal{H}}+f_1(x)\mathbb{I}_{\mathcal{H}^C}dx\\ &\geq \frac{1}{2}\int_{\mathbb{R}} \text{mín}\{f_0(x),f_1(x)\}dx\\ &=\geq \frac{1}{2}2(1-\Phi(\mu/2))=(1-\Phi(\mu/2)),\end{aligned} \tag{1.4}$$

*donde $\Phi(\mu/2))$ denota la probabilidad de que una $N(0,1)$ tome valores mayores que $\mu/2$*

**Definición 1.1.5** *Para problemas de clasificación binaria, dada cualquier distribución $D$ sobre $\mathcal{X}\times\mathcal{Y}$ el predictor de Bayes, denotado por $h_D$, se define como:*

$$h_D(x)=\begin{cases}1 & si \quad \mathbb{P}_D[y=1|x]\geq 1/2\\ 0 & si \quad \mathbb{P}_D[y=0|x]>1/2\end{cases}$$

Es fácil probar que $h_D$ es una regla óptima, en el sentido de que minimiza el riesgo:

Dada cualquier regla $h'$ y cualquier $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Observese que $h_D(x)(\mathbb{P}_D[y = 1] - \mathbb{P}_D[y = 0]) \geq h'(x)(\mathbb{P}_D[y = 1] - \mathbb{P}_D[y = 0])$. Es claro por tanto que:

$$\begin{aligned} l(h_D, x, y) &= \mathbb{P}_D[y = 1|x]\mathbb{P}_D[h_D(x) = 0] + \mathbb{P}_D[y = 0|x]\mathbb{P}_D[h_D(x) = 1] \\ &= \mathbb{P}_D[y = 0|x] + \mathbb{P}_D[h_D(x) = 0](\mathbb{P}_D[y = 1|x] - \mathbb{P}_D[y = 0|x]) \\ &\geq \mathbb{P}_D[y = 0|x] + \mathbb{P}_D[h(x) = 0](\mathbb{P}_D[y = 1|x] - \mathbb{P}_D[y = 0|x]) \\ &= l(h, x, y), \end{aligned} \tag{1.5}$$

por tanto, $L_D(h_D) \leq L_D(h')$ para cualquier regla.

Sin embargo, aunque es útil conocer la existencia de un minimizador de del riesgo, el predictor óptimo de Bayes no es práctico, pues se desconoce la distribución $D$ y por tanto dicho predictor también es desconocido.

## 1.2. Riesgo, clases de reglas y sobreajuste

Aunque la distribución subyacente es desconocida, se busca una aproximación para poder hacer inferencias sobre ella. La Ley de los grandes números asegura que cuando el tamaño de la muestra tiende a infinito:

$$\frac{1}{|S|}\sum_{i=1}^{m} l(h, (x_i, y_i)) \underset{|S|\to\infty}{\longrightarrow} \underset{(x,y)\sim D}{\mathbb{E}} l(h, (x_i, y_i)) = L_D(h).$$

Por ello se define:

**Definición 1.2.1** *(Riesgo empírico) El riesgo empírico se denota por $L_S$ y se define como:*

$$L_S = \frac{1}{|S|}\sum_{(X',Y')\in S} \mathcal{I}_{(h(X')\neq Y')}.$$

Es claro que el riesgo empírico puede ser utilizado como estimador del riesgo, de hecho, en ese principio se basa el algoritmo descrito a continuación:

**Definición 1.2.2** *(Algoritmo MRE) Dada una clase de reglas $\mathcal{H}$ se define el algoritmo de minimización del riesgo empírico (MRE de ahora en adelante) como aquel que dada una muestra $S$ devuelve $h_{MRE(S)}$, donde:*

$$h_{MRE(S)} \in \underset{h\in\mathcal{H}}{argmin} L_S(h).$$

Este algoritmo es conocido en la literatura inglesa como ERM, respondiendo a las siglas de empirical risk minimization. En la práctica este algoritmo no siempre se puede llevar a cabo, puesto que su resolución puede ser del tipo np-duro. Por ello se recuerda que durante esta memoria se llama algoritmo a cualquier función que asigne una regla a cada muestra posible, sin tener en cuenta el coste computacional que tiene calcular dicha regla.

El uso del algoritmo MRE nos da una primera justificación, en forma de ejemplo, de por qué se debe escoger una clase de reglas suficientemente pequeña, como se anticipó en la descripción del problema.

**Ejemplo 1.2.1** *Se plantea el problema de regresión polinómica (utilizando la función de pérdida cuadrática $l_2$) para ajustar los datos aleatorios obtenidos de una función polinómica de grado dos a los que se les ha añadido cierto ruido.*

*Se utiliza el algoritmo MRE para la resolución del problema y se toman tres clases de reglas distintas:*

$\mathcal{H}_1$=*Todas las funciones lineales*

$\mathcal{H}_2$=*Todos los polinomios de grado* $\leq 2$

$\mathcal{H}_9$=*Todos los polinomios de grado* $\leq 9$

*El resultado se puede observar en la Figura 1.1. En todas las imágenes la función germen en rojo aparece, los puntos que se quieren ajustar (que provienen de la función germen) en negro y el otro polinomio es el que devuelve el algoritmo MRE para cada clase de reglas. Las gráficas están colocadas de izquierda a derecha de forma creciente con respecto al tamaño de la clase utilizada.*

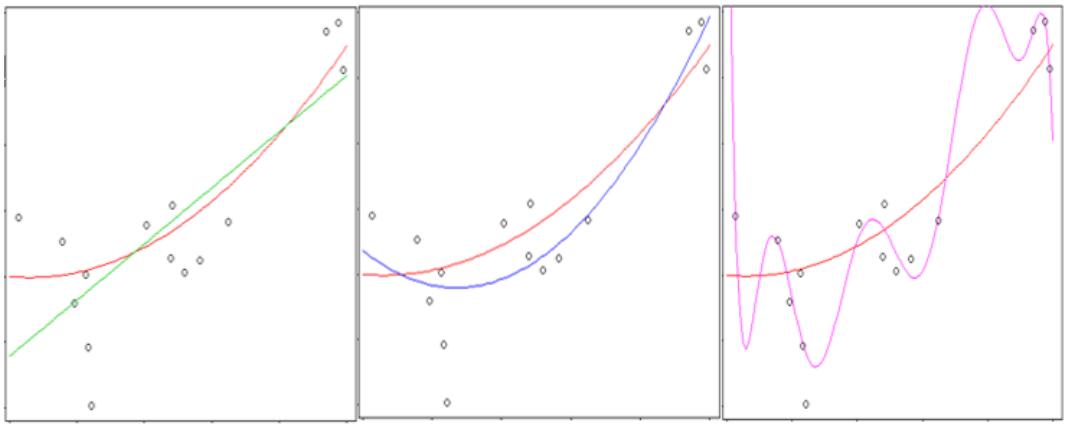

Figura 1.1: Subajuste y sobreajuste

*Se observa en la imagen que el polinomio extraído de clase más grande ajusta muy bien la muestra, sin embargo, hace un trabajo nefasto aproximando la función*

*germen. Esto significa que el polinomio de grado alto cometerá grandes errores cuando etiquete datos que no pertenezcan a la muestra y por tanto tendrá un riesgo grande.*

*Este fenómeno es conocido como sobreajuste, overfitting en la literatura inglesa. Para detectarlo se debe ocultar parte de la muestra al algoritmo MRE, para testear la regla que este algoritmo devuelve.*

*En el tercer capítulo de esta memoria se presenta una solución a este problema. De forma intuitiva esta solución consiste en considerar una gran cantidad de clases y penalizar en función de la complejidad de la clase que se elige, para que no se tomen clases excesivamente complejas.*

*El fenómeno opuesto también existe y se puede encontrar en la imagen de la izquierda. Los polinomios que se consideran en la clase $\mathcal{H}_1$ no tienen la flexibilidad necesaria como para ajustar correctamente la muestra. Por tanto la regla devuelta por el algoritmo MRE no ajustará correctamente y tendrá un riesgo grande.*

*Este fenómeno se conoce como subajuste, underfitting en la literatura inglesa. Es realmente fácil de detectar pues si está teniendo lugar el fenómeno de subajuste se tendrán necesariamente grandes errores muestrales.*

## 1.3. El paradigma de aprendizaje PAC

La muestra, $S$, es un vector aleatorio y por ello para cualquier algoritmo $A(S)$ y $L_D(A(S))$ son objetos aleatorios. No se puede esperar por tanto que el algoritmo siempre devuelva una regla 'buena'. Podría ocurrir, aunque es sumamente improbable, que la muestra estuviese formada por un único elemento de $\mathcal{X} \times \mathcal{Y}$ repetido varias veces, en ese caso el algoritmo generado por la muestra sería necesariamente 'malo'.

Teniendo en cuenta los inconvenientes recientemente mencionados, Vapnik definió formalmente la noción de aprendizaje que se introduce a continuación.

**Definición 1.3.1** *(Aprendizaje PAC) Se dice que una clase de reglas $\mathcal{H}$ es aprendible de forma probablemente aproximadamente correcta (PAC aprendible de ahora en adelante) si existe un algoritmo de aprendizaje $A$ y una función $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ tal que para todo $\epsilon, \delta \in (0,1)$, si $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, entonces para cualquier distribución $D$, con probabilidad al menos $1-\delta$ sobre la elección de la muestra $S \sim D^m$ se tiene:*

$$L_D(A(S)) \leq \min_{h \in \mathcal{H}} L_D(h) + \epsilon.$$

Se busca por tanto un tamaño muestral que asegure que existe un algoritmo que, con la seguridad deseada, $1-\delta$, devuelva una regla casi tan precisa ($\epsilon$) como la óptima en la clase. Esta noción se conoce como *PAC learnability* en la

literatura inglesa, respondiendo a las siglas de Probably Approximately Correct learnability. Se llamará complejidad estadística de la clase $\mathcal{H}$ para el aprendizaje PAC a la mínima función $m_{\mathcal{H}}$ válida para la definición previa.

En la descripción del problema se ha mencionado el concepto de muestra suficientemente representativa, este se formaliza a continuación:

**Definición 1.3.2** *($\epsilon$-representatividad) Se dice que una muestra es $\epsilon$-representativa si:*

$$|L_D(h) - L_S(h)| \leq \epsilon \text{ para todo } h \in \mathcal{H}.$$

Esta noción incita a definir la propiedad de la convergencia uniforme para las clases de reglas.

**Definición 1.3.3** *(Convergencia Uniforme)Se dice que una clase $\mathcal{H}$ tiene la propiedad de la Convergencia Uniforme si existe un algoritmo de aprendizaje $A$ y una función $m_{\mathcal{H}}^{CU} : (0,1)^2 \to \mathbb{N}$ tal que para todo $\epsilon, \delta \in (0,1)$ si $m \geq m_{\mathcal{H}}^{CU}(\epsilon, \delta)$, entonces para cualquier distribución $D$ con probabilidad al menos $1-\delta$ sobre la elección de la muestra $S \sim D^m$ se tiene:*

$$|L_D(h) - L_S(h)| \leq \epsilon \text{ para todo } h \in \mathcal{H}.$$

Se llamará complejidad estadística de la clase $\mathcal{H}$ para la convergencia uniforme a la mínima función $m_{\mathcal{H}}^{CU}$ válida para la definición previa. En el tercer capítulo de la memoria la propiedad de la convergencia uniforme se probará equivalente al aprendizaje PAC, esto será muy útil para alcanzar resultados sobre este tipo aprendizaje.

No pasa desapercibida la relación entre las nociones de convergencia uniforme y la convergencia uniforme en probabilidad de $L_{S_m}(h)$ hacia $L_D(h)$ en $\mathcal{H}$ que se da cuando:

$$\mathbb{P}(\sup_{h \in \mathcal{H}} |L_{S_m}(h) - L_D(h)| \geq \epsilon) \to 0 \text{ cuando } m \to \infty.$$

Sin embargo, los dos conceptos no son exactamente equivalentes pues para poder hablar de la probabilidad anterior debería ocurrir que $\sup_{h \in \mathcal{H}} |L_{S_m}(h) - L_D(h)|$ fuese medible, en cuyo caso sí se daría la equivalencia. Se observa por tanto que la definición de convergencia uniforme que vamos a manejar evade hábilmente los problemas de medibilidad.

Como primer paso hacia la demostración de la equivalencia entre la noción de aprendizaje PAC y la noción de convergencia uniforme se prueba:

**Proposición 1.3.1** *Si $\mathcal{H}$ tiene la propiedad de la convergencia uniforme entonces $\mathcal{H}$ es PAC aprendible. Además:*

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{CU}(\epsilon/2, \delta). \tag{1.6}$$

**Demostración.** Dada la muestra $S$ y denotando por $h^* \in argmin_{h' \in \mathcal{H}} L_D(h')$ a la regla que devuelve el algoritmo MRE se tiene:

$$\begin{aligned} L_D(A(S)) - L_D(h^*) &\leq L_D(A(S)) - L_S(h^*) + L_S(h^*) - L_D(h^*) \\ &\leq L_D(A(S)) - L_S(A(S)) + L_S(h^*) - L_D(h^*) \\ &\leq |L_D(A(S)) - L_S(A(S))| + |L_S(h^*) - L_D(h^*)| \\ &\leq \sup_{h \in \mathcal{H}} 2|L_S(h) - L_D(h)|. \end{aligned} \tag{1.7}$$

Si $\mathcal{H}$ tiene la propiedad de la convergencia uniforme, entonces existe $m_{\mathcal{H}}^{CU}(\epsilon/2, \delta)$ tal que con probabilidad al menos $1 - \delta$ se tiene $|L_D(h) - L_S(h)| \leq \epsilon/2$. Y por tanto por la desigualdad previa $L_D(A(S)) - L_D(h^*) \leq \epsilon$ con probabilidad al menos $1 - \delta$. Por lo que:

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{CU}(\epsilon/2, \delta). \tag{1.8}$$

□

**Corolario 1.3.2** *El algoritmo MRE es válido para conseguir las garantías necesarias para el aprendizaje PAC (resp. la convergencia uniforme) en cualquier clase de reglas aprendible PAC (resp. con la propiedad de la convergencia uniforme).*

**Demostración.** Se obtiene directamente de la demostración previa. □

Sería deseable tener una caracterización de las clases de reglas que son aprendibles PAC, en busca de la misma se probará que toda clase de reglas finita es aprendible PAC.

Para alcanzar este resultado es suficiente la Ley de los grandes números, sin embargo la desigualdad de Hoeffding que se enuncia a continuación no solo prueba el resultado, sino que además proporciona una cota de $m_{\mathcal{H}}(\epsilon, \delta)$ para toda $\mathcal{H}$ finita.

**Lema 1.3.3** *(Hoeffding)Siendo $\theta_1, ..., \theta_m$ una secuencia de variables aleatorias i.i.d., donde para todo $i$, $\mathbb{E}[\theta_i] = \mu$ y $\mathbb{P}[a \leq \theta_i \leq b] = 1$. Para todo $\epsilon > 0$:*

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m}(\theta_i - \mu)\right| > \epsilon\right] \leq 2exp(-2m\epsilon^2/(b-a)^2).$$

**Teorema 1.3.4** *Toda clase finita de reglas es aprendible PAC.*

**Demostración.** Se prueba que toda clase finita de reglas tiene la propiedad de la convergencia uniforme y por tanto gracias a la Proposición 1.3.1 se tendrá que la clase es aprendible PAC.

Fijados $\epsilon > 0$ y $\delta \in (0, 1)$ se debe probar que existe un $m$ tal que:

$$\mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| \leq \epsilon \ \forall h \in \mathcal{H}] \geq 1 - \delta,$$

o lo que es lo mismo, que se cumpla con probabilidad menor que $\delta$:

$$\mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| \geq \epsilon$$

para algún $h \in \mathcal{H}$.

A su vez lo anterior se cumple sí y solo sí:

$$\prod_{h\in\mathcal{H}} \mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| \geq \epsilon] \leq \delta.$$

Que gracias a la propiedad de que la probabilidad de una unión es menor que la suma de las probabilidades se cumple si:

$$\sum_{h\in\mathcal{H}} \mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| \geq \epsilon] \leq \delta,$$

donde es fácil comprobar que para todo $h \in \mathcal{H}$, $\mathbb{E}[L_S(h)] = L_D(h)$ y $\mathbb{P}[0 \leq L_S(h) \leq 1] = 1$, por lo que para cada $h \in \mathcal{H}$ aplicando la desigualdad de Hoeffding 1.3.3:

$$\mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| \geq \epsilon] \leq 2exp(-2m\epsilon^2).$$

Por lo que, para que se cumpla la propiedad de la convergencia uniforme, $m$ debe cumplir:

$$\sum_{h\in\mathcal{H}} 2exp(-2m\epsilon^2) = 2|\mathcal{H}|exp(-2m\epsilon^2) \leq \delta.$$

Tomando por tanto:

$$m = \left\lceil \frac{log\frac{\delta}{2|\mathcal{H}|}}{2\epsilon^2} \right\rceil, \tag{1.9}$$

se tiene que se cumple la convergencia uniforme para toda clase de reglas finita. □

**Corolario 1.3.5** *Para toda clase de reglas $\mathcal{H}$ finita:*

$$m_{\mathcal{H}}^{CU}(\epsilon, \delta) \leq \left\lceil \frac{log\frac{\delta}{2|\mathcal{H}|}}{2\epsilon^2} \right\rceil,$$

y:

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{CU}(\epsilon/2, \delta) \leq \left\lceil \frac{2log\frac{\delta}{2|\mathcal{H}|}}{\epsilon^2} \right\rceil .$$

**Demostración.** Se tiene el resultado de forma inmediata de (1.9) y (1.6).

□

Cabe destacar que muchas clases infinitas de reglas, al ser representadas en el ordenador, por la representación de punto flotante de 64 bits, por ejemplo, se convierten en clases finitas a las que se puede aplicar la cota previa. Sin embargo la cota anterior depende del cardinal de la clase y probablemente la discretización conduciría a cotas pobres.

# Capítulo 2

# El Teorema Fundamental del aprendizaje estadístico

Se ha visto durante la sección anterior que para que una clase $\mathcal{H}$ sea PAC aprendible es suficiente que sea finita, pero esto no es una condición necesaria. Se busca durante este capítulo una caracterización de las clases PAC aprendibles.

## 2.1. Teorema de la Imposibilidad

En el Corolario 1.3.5 se tiene una cota del tamaño de la muestra que se necesita para alcanzar cierta precisión, se busca ahora qué precisiones son inalcanzables dado cierto tamaño de muestra. Para llegar a este resultado de forma clara se definen:

**Definición 2.1.1** *(Restricción de $\mathcal{H}$ a $C$) Dada una clase de reglas $\mathcal{H}$ y un conjunto $C = \{c_1, ..., c_m\}$ se define la* restricción de $\mathcal{H}$ a $C$ *como el conjunto de regla de $C$ a $\{0,1\}$*

$$\mathcal{H}_C = \{(h(c_1), ..., h(c_m) : h \in \mathcal{H}\}.$$

**Definición 2.1.2** *(Fragmentar) Dada clase de reglas $\mathcal{H}$ y un conjunto $C$, diremos que $H$ fragmenta a $C$ cuando la restricción de $\mathcal{H}$ a $C$ es el conjunto de todas las funciones posibles de $C$ a $\{0, 1\}$, es decir, cuando $H_C = 2^{|C|}$.*

Para proceder con la demostración del Teorema principal de esta sección se debe probar el siguiente resultado auxiliar:

**Lema 2.1.1** *Dada una variable aleatoria $X$ con $\mathbb{P}[0 \leq X \leq 1] = 1$, entonces $\delta = \frac{1}{1-\epsilon}(E - \epsilon)$ es la mínima $\delta$ posible con $\mathbb{P}[X \leq \epsilon] = 1 - \delta$ y con $\mathbb{E}[X] \geq E$.*

**Demostración.** Es claro que se busca la $\delta$ tal que si:

$$\begin{aligned} \mathbb{P}[X = 1] &= \delta \\ \mathbb{P}[X = \epsilon] &= 1 - \delta, \end{aligned} \tag{2.1}$$

entonces $\mathbb{E}[X] = E$

Se debe cumplir por tanto:

$$E = \delta + \epsilon(1 - \delta) = \epsilon + \delta(1 - \epsilon).$$

Lo que implica que: $\delta = \frac{1}{1-\epsilon}(E - \epsilon)$ como se pretendía probar.

□

Ahora se procede con el enunciado y la demostración del resultado principal de la sección.

**Teorema 2.1.2** *(Teorema de la Imposibilidad)Dado un espacio de atributos $\mathcal{X}$ y dada una muestra $S$ con $|S| \leq |\mathcal{X}|/n$. Para cualquier algoritmo de aprendizaje $A$ que escoge una regla de una clase $\mathcal{H}$ que fragmenta a un conjunto $C$ con $|S| \leq |C|/n$ se tiene que:*

*Dados $\epsilon, \delta \in (0, 1)$ cumpliendo $\delta < \left(\frac{n-1}{2n} - \epsilon\right) \frac{1}{1-\epsilon}$ existe una distribución $D$ tal que con probabilidad $1 - \delta$ sobre la elección de la muestra se tiene:*

$$L_D(A(S)) > \epsilon.$$

**Demostración.** Sea m el tamaño de la muestra $S$ y dado $C = [c_1, ..., c_{nm}]$ un subconjunto de $\mathcal{X}$ de tamaño $nm$ se denotan por $f_1, ..., f_T$ las $T = 2^{nm}$ funciones posibles de $C$ a $\{0, 1\}$. Para cada una de estas funciones se diseña una distribución $D_i$ sobre $C \times \{0, 1\}$ definida por:

$$\mathbb{P}_{D_i}[\{(x, y)\}] = \begin{cases} 1/|C| & si \quad y = f_i(x) \\ \\ 0 & si \quad y \neq f_i(x) \end{cases} \tag{2.2}$$

Se probará que para todo algoritmo de aprendizaje $A$, existe una distribución $D_i$ tal que:

$$\underset{S \sim {D_i}^m}{\mathbb{E}}[L_{D_i}(A(S))] \geq \frac{n-1}{2n}. \tag{2.3}$$

Y como $L_{D_i}(A(S)) \leq 1$ con probabilidad 1, aplicando el Lema 2.1.1 se tiene que $\delta = \left(\frac{n-1}{2n} - \epsilon\right) \frac{1}{1-\epsilon}$ es el mínimo $\delta$ para el que es posible que con probabilidad $1 - \delta$ sobre la elección de la muestra:

$$L_{D_i}(A(S)) \leq \epsilon.$$

Esto probaría el Teorema de Imposibilidad. Por lo que solo queda probar (2.3).

Fijada un distribución $D_i$ se denotan $S_1^i, ..., S_k^i$ a las $k = (nm)^m$ muestras posibles de $m$ elementos de $C$. Todas estas muestras son igualmente probables bajo la distribución $D_i$, por lo que:

$$\mathop{\mathbb{E}}_{S\sim D^i}[L_{D_i}(A(S))] = \frac{1}{k}\sum_{j=1}^{k}[L_{D_i}(A(S_j^i))]. \tag{2.4}$$

Utilizando propiedades elementales:

$$\begin{aligned}\max_{i\in[T]}[L_{D_i}(A(S))] &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{k}\sum_{j=1}^{k}L_{D_i}(A(S_j^i)) \\ &= \frac{1}{k}\sum_{j=1}^{k}\frac{1}{T}\sum_{i=1}^{T}L_{D_i}(A(S_j^i)) \\ &\geq \min_{j\in[k]}\frac{1}{T}\sum_{i=1}^{T}(A(S_j^i))\end{aligned} \tag{2.5}$$

.

Fijado un $j \in [k]$ se denota por $x_1, .., x_m$ a los elementos de $S_j$ y por $v_1, .., v_p$ a los elementos del espacio de atributos que no están contenidos en $S_j$. Por el tamaño de $C$, $nm$, es claro que $m = \frac{|C|}{n} \leq \frac{p}{n-1}$. Por ello se cumple todo $i$ y toda regla $h : \mathcal{X} \to \{0, 1\}$:

$$\begin{aligned}L_D(h) &\geq \frac{1}{nm}\sum_{r=1}^{p}\mathbb{I}_{h(v_r)\neq f_i(v_r)} \\ &\geq \frac{n-1}{np}\sum_{r=1}^{p}\mathbb{I}_{h(v_r)\neq f_i(v_r)},\end{aligned} \tag{2.6}$$

por tanto:

$$\begin{aligned}\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}[L_{D_i}(A(S_j^i))] &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{n-1}{np}\sum_{r=1}^{p}\mathbb{I}_{A(S_j^i)(v_r)\neq f_i(v_r)} \\ &= \frac{n-1}{np}\sum_{r=1}^{p}\frac{1}{T}\sum_{i=1}^{T}\mathbb{I}_{A(S_j^i)(v_r)\neq f_i(v_r)} \\ &= \frac{n-1}{n}\min_{r\in[p]}\frac{1}{T}\sum_{i=1}^{T}\mathbb{I}_{A(S_j^i)(v_r)\neq f_i(v_r)}\end{aligned} \tag{2.7}$$

Fijada $r \in [p]$, las funciones $f_1, ..., f_T$ se pueden emparejar en $T/2$ pares $(f_i, f_{i'})$ de forma que:

$$\begin{cases} f_i(c) = f_{i'}(c) & si \quad c \neq v_r \\ f_i(c) \neq f_{i'}(c) & si \quad c = v_r \end{cases}$$

Estas parejas cumplen:

$$\mathbb{I}_{A(S_j^i)(v_r) \neq f_i(v_r)} + \mathbb{I}_{A(S_j^{i'})(v_r) \neq f_{i'}(v_r)} = 1,$$

de donde se deduce:

$$\mathbb{I}_{A(S_j^i)(v_r) \neq f_i(v_r)} = \frac{1}{2} \tag{2.8}$$

Combinando (2.4),(2.5),(2.7) y (2.8) se obtiene (2.3) concluyéndose así la prueba. □

La prueba de este resultado ha sido extraída de [3] y generalizada para el caso no realizable y para los distintos ratios entre los tamaños de las muestras y los tamaños de los espacios de atributos.

En oposición a cuando se restringe la clase de reglas y se toma una de las reglas que mejor se ajusta a la muestra. Al considerar todas las reglas posibles, el conocimiento aportado por la muestra deja abiertas todas las reglas posibles para el resto de atributos, por lo que estos se etiquetarán *a ciegas*.

Evidentemente reducir la clase de reglas aleatoriamente no induce ninguna mejora, por lo que se debe tener necesariamente algún conocimiento previo sobre el problema para reducir la clase de reglas y poder resolverlo.

## 2.2. La dimensión de Vapnik-Chervonenkis

En busca de una caracterización correcta para las clases aprendibles se define la dimensión Vapnik-Chervonenkis, esta definición viene motivada por el Teorema de la Imposibilidad 2.1.2 y por su demostración. En la prueba del resultado se utiliza que existe un subconjunto $C$ del espacio de atributos $\mathcal{X}$ al menos $n$ veces más grande que la muestra con la propiedad de que la clase de reglas $\mathcal{H}$ contenga todas las funciones posibles de $C$ a $\{0, 1\}$.

**Definición 2.2.1** *(Dimensión Vapnik-Chervonenkis) La Dimensión-VC de $\mathcal{H}$, que se denota por VCdim($\mathcal{H}$), se define como el supremo de los tamaños de los conjuntos que son fragmentados por $\mathcal{H}$.*

Esta definición nos permite dar un enunciado más conciso para el Teorema de la Imposibilidad:

**Teorema 2.2.1** *(Teorema de la Imposibilidad)Dada una clase de reglas $\mathcal{H}$ con $DimVC(\mathcal{H}) = d$ y dada una muestra $S$ con $|S| \leq d/n$, para cualquier algoritmo de aprendizaje $A$ se tiene que:*

*Dados $\epsilon, \delta \in (0,1)$ cumpliendo $\delta < \left(\frac{n-1}{2n} - \epsilon\right)\frac{1}{1-\epsilon}$ existe una distribución $D$ tal que con probabilidad $1-\delta$ sobre la elección de la muestra se tiene:*

$$L_D(A(S)) > \epsilon.$$

Esta formulación nos permite alcanzar resultados muy interesantes:

**Corolario 2.2.2** *Dada una clase de reglas $\mathcal{H}$ con $DimVC(\mathcal{H}) = \infty$, para cualquier tamaño de muestra y para cualquier algoritmo de aprendizaje $A$ se tiene que:*

*Dados $\epsilon, \delta \in (0,1)$ cumpliendo $\delta < \left(\frac{1}{2} - \epsilon\right)\frac{1}{1-\epsilon}$ existe una distribución $D$ tal que con probabilidad $1-\delta$ sobre la elección de la muestra $S$ se tiene:*

$$L_D(A(S)) > \epsilon.$$

**Demostración.** Fijado el tamaño muestral $m$ se puede encontrar un $C_n$, de tamaño $nm$, tal que $\mathcal{H}$ fragmenta a $C_n$, por lo que:

Para todo $n$, dados $\epsilon, \delta \in (0,1)$ cumpliendo $\delta < \left(\frac{n-1}{2n} - \epsilon\right)\frac{1}{1-\epsilon}$ existe una distribución $D$ tal que con probabilidad $1-\delta$ sobre la elección de la muestra $S$:

$$L_D(A(S)) > \epsilon.$$

Se termina la demostración observando que para todo $\epsilon, \delta \in (0,1)$ cumpliendo $\delta < \left(\frac{1}{2} - \epsilon\right)\frac{1}{1-\epsilon}$, se puede encontrar un $n$ tal que $\delta < \left(\frac{n-1}{2n} - \epsilon\right)\frac{1}{1-\epsilon}$.

□

**Corolario 2.2.3** *Dada una clase de reglas $\mathcal{H}$ y dados $\epsilon, \delta \in (0,1)$ cumpliendo $\delta < \left(\frac{1}{2} - \epsilon\right)\frac{1}{1-\epsilon}$, si para algún tamaño de muestra, algún algoritmo de aprendizaje $A$ asegura que para toda distribución $D$ con probabilidad $1-\delta$ sobre la elección de la muestra $S$ se tiene:*

$$L_D(A(S)) \leq \epsilon,$$

*entonces $DimCV(\mathcal{H}) < \infty$*

**Demostración.** Es el contrarrecíproco del Corolario 2.2.2 □

Se termina la sección calculando la dimensión-VC de la clase de reglas del siguiente ejemplo.

**Ejercicio 2.2.4** *Sea $\mathcal{H}^d_{rec}$ la clase de rectángulos alineados con los ejes en $\mathbb{R}^d$, pruebese que $VCdim(\mathcal{H}^d_{rec}) = 2d$ que a partir de ahora denotaremos por $\mathcal{H}^d_{rec}$.*

Para probar que una clase de reglas $\mathcal{H}$ tiene dimensión $2d$, se debe probar que existe un conjunto $C$ de $2d$ puntos que puede ser fragmentado por $\mathcal{H}$. Y que ningún conjunto $C'$ de $2d+1$ puntos puede ser fragmentado por $\mathcal{H}$. Se procede a continuación a realizar el ejercicio anterior utilizando esta notación.

Se define $C = \{\pm e_i : i \leq d\}$ donde los $e_i$ son los elementos de la base canónica de $\mathbb{R}^d$. Es claro que $C$ contiene $2d$ puntos y además, dado cualquier subconjunto $S \subset C$ se puede diseñar un rectángulo $R \in \mathcal{H}^d_{rec}$ tal que $C \cap R = S$ de la siguiente forma:

$$R = \prod_{i=1}^{d} [a_i, b_i],$$

donde

$$a_i = \begin{cases} 1 & si \quad -e_i \in S \\ \\ 0 & si \quad -e_i \notin S \end{cases}$$

y donde

$$b_i = \begin{cases} 1 & si \quad e_i \in S \\ \\ 0 & si \quad e_i \notin S \end{cases}$$

por lo que $\mathcal{H}$ fragmenta a $C$ y por tanto $VCdim(\mathcal{H}) \geq 2d$.

Para la segunda parte del ejercicio, dado un conjunto de $2d+1$ puntos $C' = \{p^1, ..., p^{2d+1}\}$, se selecciona en cada una de las d dimensiones dos puntos, $p^{i-}$ y $p^{i+}$ donde $i_- \in argmin_{1 \leq n \leq 2d+1}\{p^n_i\}$ y donde $i_+ \in argmax_{1 \leq n \leq 2d+1}\{p^n_i\}$. A lo sumo, el conjunto $P = \{p^{i\pm} : i \leq d\}$ tendrá $2d$ puntos, sin embargo, cualquier rectángulo $R \in \mathcal{H}^d_{rec}$ que contenga a $P$ contendrá a los $2d+1$ puntos de $C'$. Puesto que $\forall n, i \; p^{i-} \leq p^n_i \leq p^{i+}$, así que ningún conjunto de $2d+1$ puntos es fragmentable por $\mathcal{H}$ y por tanto $VCdim(\mathcal{H}^d_{rec}) = 2d$.

## 2.3. Función de crecimiento y Lema de Sauer

En esta sección se prueba el resultado principal del capítulo, el Teorema Fundamental del Aprendizaje Estadístico, que proporciona una caracterización del aprendizaje PAC. Para completar la prueba de este resultado se demostrará que si una clase tiene dimensión-VC finita entonces tiene la propiedad de la convergencia uniforme. Se comienza por introducir el siguiente concepto:

**Definición 2.3.1** *(Función de crecimiento). Sea $\mathcal{H}$ una clase de reglas. Se define la función de crecimiento de $\mathcal{H}$, denotada por $\tau_H(m) : \mathbb{N} \to \mathbb{N}$, como:*

$$\tau_H(m) = \sup_{C \subset X : |C| = m} |H_C|.$$

$\tau_H(m)$ es por tanto el número de funciones distintas de un conjunto $C$ de tamaño $m$ a $\{0,1\}$ que se pueden obtener restringiendo $\mathcal{H}$ a $C$. Es claro que $\tau_{\mathcal{H}}(m) = 2^m$ si $m \leq VCdim(\mathcal{H})$, pero gracias al Lema de Sauer que se enunciará a continuación se sabe que $\tau_{\mathcal{H}}(m)$ crece de forma polinómica cuando $m > VCdim(\mathcal{H}) + 1$, lo cual resultará realmente útil para encontrar mejores cotas de $m_{\mathcal{H}}^{CU}$.

**Lema 2.3.1** *(Lema de Sauer) Sea $\mathcal{H}$ una clase de reglas con $VCdim(\mathcal{H}) \leq d < \infty$. Entonces $\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}$ para todo $m$. Además si $m > d+1$ se tiene $\tau_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d$.*

**Demostración.** Para cada $C = c_1, ..., c_m \subset \mathcal{X}$, se tiene:

$$|\{B \subset C : \mathcal{H} fragmenta a B\}| \leq |\{B \subset C\}| = \sum_{i=0}^{d} \binom{m}{i},$$

por lo que demostrar:

$$|\mathcal{H}_C| \leq |\{B \subset C : \mathcal{H} fragmenta a B\}|. \tag{2.9}$$

Es suficiente para probar:

$$\tau_{\mathcal{H}}(m) = \sup_{C \subset \mathcal{X} : |C| = m} |\mathcal{H}_C| \leq \sum_{i=0}^{d} \binom{m}{i}.$$

Además cuando $m > d+1$, $\sum_{i=0}^{d} \binom{m}{i} \leq \dfrac{em}{d}$, por lo que (2.9) prueba el Lema de Sauer.

(2.9) se prueba por inducción sobre m:

• Si $m = 1$ se dará siempre la desigualdad, puesto que el conjunto vacío siempre es fragmentado por $\mathcal{H}$ y cuando $|\mathcal{H}_C| = 2^m = 2$, también el conjunto con el único elemento de $C$ es fragmentado por $\mathcal{H}$.

• Se supone cierta la desigualdad (2.9) para conjuntos de tamaño $k < m$ y se trata de probar que es cierta para $k = m$:

Para ello dados $\mathcal{H}$ y $C = \{c_1, ..., c_m\}$, se denota $C' = \{c_2, ..., c_m\}$ y se definen los conjuntos:

$$Y_0 = \{(y_2, ..., y_m) : (0, y_2, ..., y_m) \in \mathcal{H}_C o (1, y_2, ..., y_m) \in \mathcal{H}_C\}.$$

$$Y_1 = \{(y_2, ..., y_m) : (0, y_2, ..., y_m) \in \mathcal{H}_C y (1, y_2, ..., y_m) \in \mathcal{H}_C\}.$$

Es claro que $|Y_0| + |Y_1| = |\mathcal{H}_C|$.
Además como $Y_0 = \mathcal{H}_{C'}$, utilizando la hipótesis de inducción:

$$\begin{aligned} |Y_0| = |\mathcal{H}_{C'}| &\leq |\{B \subset C' : \mathcal{H} \ fragmenta \ a \ B\}| \\ &= |\{B \subset C' : c_1 \notin B \ y \ \mathcal{H} \ fragmenta \ a \ B\}|. \end{aligned} \tag{2.10}$$

A continuación se define:

$$\mathcal{H}' = h \in \mathcal{H} : \exists h' \in \mathcal{H} \ tal \ que \ (1 - h'(c_1), h'(c_2), ..., h'(c_m)) = (h(c_1), h(c_2), ..., h(c_m)).$$

Es claro que $\mathcal{H}' \subset \mathcal{H}$ y que $\mathcal{H}'$ fragmenta a $B$ sí y solo sí fragmenta a $B \cup \{c_1\}$.
Usando además que $Y_1 = \mathcal{H}'_{C'}$ y la hipótesis de inducción:

$$\begin{aligned} |Y_1| = |\mathcal{H}'_{C'}| &\leq |\{B \subset C' : \mathcal{H}' \ fragmenta \ a \ B\}| \\ &= |\{B \subset C' : \mathcal{H}' \ fragmenta \ a \ B \cup \{c_1\}\}| \\ &= |\{B \subset C : c_1 \in B \ y \ \mathcal{H}' \ fragmenta \ a \ B\}| \\ &\leq |\{B \subset C : c_1 \in B \ y \ \mathcal{H} \ fragmenta \ a \ B\}| \end{aligned} \tag{2.11}$$

Finalmente teniendo todo lo anterior en cuenta:

$$\begin{aligned} |\mathcal{H}_C| &= |Y_0| + |Y_1| \\ &\leq |\{B \subset C : c_1 \notin B \ y \ \mathcal{H} \ fragmenta \ a \ B\}| + |\{B \subset C : c_1 \in B \ y \ \mathcal{H} \ fragmenta \ a \ B\}| \\ &= |\{B \subset C : \mathcal{H} \ fragmenta \ a \ B\}| \end{aligned} \tag{2.12}$$

Lo que concluye la prueba.

□

**Definición 2.3.2** *Se dice que la variable $\sigma$ es Rademacher si:*

$$\mathbb{P}[\sigma = 1] = \mathbb{P}[\sigma = -1] = 1/2$$

*. Se denotará la distribución de Rademacher por $U_{\pm}$.*

Se encadenan a continuación una serie de enunciados y pruebas que culminan con la demostración del Teorema Fundamental del Aprendizaje Estadístico.

**Nota 2.3.2** *Cuando $A_1$ y $A_2$ son variables i.i.d. con distribución $B$ y $\sigma$ Rademacher independiente de $A_1$ y $A_2$:*

$$A_1 - A_2 \text{ es igual en distribucion que } \sigma(A_1 - A_2).$$

Es claro que:

$$\mathbb{P}[A_1 - A_2 > \epsilon] = \mathbb{P}[A_2 - A_1 > \epsilon] = \mathbb{P}[A_1 - A_2 < -\epsilon],$$

por tanto:

$$\begin{aligned}\mathbb{P}[\sigma(A_1 - A_2) > \epsilon] &= \mathbb{P}[\sigma = 1]\mathbb{P}A[_1 - A_2 > \epsilon] + \mathbb{P}[\sigma = -1]\mathbb{P}[A_1 - A_2 < -\epsilon] \\ &= 1/2\mathbb{P}[A_1 - A_2 > \epsilon] + 1/2\mathbb{P}[A_1 - A_2 < -\epsilon] \\ &= \mathbb{P}[A_1 - A_2 > \epsilon].\end{aligned} \tag{2.13}$$

**Lema 2.3.3** *Sea $a \geq 2$. Entonces $x \geq 2alog(a)$ implica que $x \geq alog(x)$.*

**Demostración.** Definiendo $f(x) = x - alog(x)$, véase que si $x \geq 2alog(a)$ entonces $f(2alog(a)) \geq 0$ y $f'(x) \geq 0$. Esto sería suficiente para probar que $f(x) = x - alog(x) \geq 0$.

Puesto que $a \geq 2$, de $x \geq 2alog(a)$ se deduce $x \geq a$. Se tiene por tanto:

$$f'(x) = 1 - \frac{a}{x} \geq 0,$$

además:

$$\begin{aligned}f(2alog(a)) &= 2alog(a) - alog(2alog(a)) \\ &= 2alog(a) - alog(a) - alog(2log(a)) \\ &= alog(a) - alog(2log(a)) \geq 0\end{aligned} \tag{2.14}$$

porque $a \geq 2log(a)$ para todo $a \geq 2$.

□

**Lema 2.3.4** *Sea $a \geq 1$ y $b > 0$. Entonces $x \geq 4alog(2a) + 2b$ implica que $x \geq alog(x) + b$.*

**Demostración.** Basta ver que $x \geq 4alog(2a) + 2b$ implica que $x \geq 2alog(x)$ y que $x \geq 2b$.

Puesto que $a \geq 1$ la segunda desigualdad es inmediata de $x \geq 4alog(2a) + 2b$.

Además puesto que $b > 0$ se deduce de $x \geq 4alog(2a) + 2b$ que $x \geq 4alog(2a)$. Y la desigualdad $x \geq 2alog(x)$ viene dada claramente por el Lema previo. □

**Lema 2.3.5** *Sea $X$ una variable aleatoria centrada, es decir con $\mathbb{E}[X] = 0$, que toma valores en el intervalo $[a, b]$, entonces para toda $s > 0$:*

$$\mathbb{E}[e^{sX}] \leq e^{\frac{s^2(b-a)^2}{8}} \tag{2.15}$$

**Demostración.** Dado que la exponencial es una función convexa y que $a \leq x \leq b$ se tiene:

$$e^{sX} \leq \frac{b-x}{b-a}e^{sa} + \frac{x-a}{b-a}e^{sb} \tag{2.16}$$

Usando que $\mathbb{E}[X] = 0$ y escribiendo $p = -\frac{a}{b-a}$, y equivalentemente $1-p = \frac{b}{b-a}$, se deduce:

$$\begin{aligned}\mathbb{E}[e^{sX}] &\leq \frac{be^{sa} - ae^{sb} + (e^{sa} - e^{sb})\mathbb{E}[x]}{b-a} = (1-p)e^{sa} + pe^{sb} \\ &= (1-p)e^{-sp(b-a)} + pe^{s(1-p)(b-a)} = \left[(1-p) + pe^{s(b-a)}\right] e^{-sp(b-a)}\end{aligned} \tag{2.17}$$

Tomando $u = s(b-a)$ y $\phi(u) = -pu + log(1 - p + pe^u)$ la cota previa se reescribe como

$$\mathbb{E}(e^{sX}) \leq e^{\phi(u)} \tag{2.18}$$

Se calculan los valores de las derivadas de $\phi$ para aplicar el desarrollo de Taylor en 0:

$$\begin{aligned}&\phi(0) = 0 \\ &\phi'(u) = -p + \frac{pe^u}{1-p+pe^u} \\ &\phi'(0) = 0 \\ &\phi''(u) = \frac{p(1-p)e^u}{(1-p+pe^u)^2} \leq \frac{1}{4}\end{aligned} \tag{2.19}$$

Por lo que existe un $\theta \in [0, u]$ con:

$$\phi(u) = \phi(0) + u\phi'(0) + \frac{u^2}{2}\phi''(\theta) \leq \frac{u^2}{8} = \frac{s^2(b-a)^2}{8} \tag{2.20}$$

Finalmente combinando (2.18) y (2.20) se obtiene la prueba del Lema.

□

**Proposición 2.3.6** *Sean* $S_j = \frac{1}{m}\sum_{i=1}^{m} X_{ij}$ *donde* $\{X_{ij}\}_{i=1,\dots,m|j=1,\dots,T}$ *son variables aleatorias independientes y centradas que toman valores en un intervalo de longuitud* $(b-a)$*, entonces:*

$$\mathbb{E}\left[\sup_{1\leq j\leq T} S_j\right] \leq (b-a)\sqrt{\frac{log(T)}{2m}} \tag{2.21}$$

**Demostración.**

El Lema 2.3.5 asegura que para todo $s > 0$:

$$\mathbb{E}[e^{\frac{s}{m}X_{ij}}] \leq e^{\frac{s^2(b-a)^2}{8m^2}} \text{ para todo } i \leq m, j \leq T \tag{2.22}$$

Gracias a la desigualdad de Jensen:

$$\begin{aligned} e^{s\mathbb{E}[\sup_{1\leq j\leq T} S_j]} &\leq \mathbb{E}[e^{\sup_{1\leq j\leq T} sS_j}] = \mathbb{E}[\sup_{1\leq j\leq T} e^{sS_j}] \\ &\leq \sum_{j=1}^{T} \mathbb{E}[e^{sS_j}] = \sum_{j=1}^{T} \mathbb{E}\left[e^{\sum_{1\leq i\leq m}\frac{s}{m}X_{ij}}\right] \\ &= \sum_{j=1}^{T} \mathbb{E}\left[\prod_{i=1}^{m} e^{\frac{s}{m}X_{ij}}\right] = \sum_{j=1}^{T}\prod_{i=1}^{m} \mathbb{E}\left[e^{\frac{s}{m}X_{ij}}\right] \\ &\leq \sum_{j=1}^{T}\prod_{i=1}^{m} e^{\frac{s^2(b-a)^2}{8m^2}} = Te^{\frac{s^2(b-a)^2}{8m}} \\ &= e^{log(T)+\frac{s^2(b-a)^2}{8m}} \end{aligned} \tag{2.23}$$

De la ecuación anterior se deduce:

$$\mathbb{E}\left[\sup_{1\leq j\leq T} S_j\right] \leq \frac{log(T)}{s} + \frac{s(b-a)^2}{8m} \tag{2.24}$$

Como esta cota es válida para todo $s > 0$ se toma $s = \frac{1}{b-a}\sqrt{8log(T)m}$ que es donde el lado derecho de (2.24) alcanza el mínimo:

$$\mathbb{E}\left[\sup_{1\leq j\leq T} S_j\right] \leq (b-a)\sqrt{\frac{log(T)}{2m}} \tag{2.25}$$

□

**Corolario 2.3.7** *Sean* $\{X_{ij}\}_{i=1,\dots,m|j=1,\dots,T}$ *variables aleatorias independientes y centradas que toman valores en* $[a,b]$*, definiendo* $S_j = \frac{1}{m}\sum_{i=1}^{m} X_{ij}$ *entonces:*

$$\mathbb{E}\left[\sup_{1\leq j\leq T} |S_j|\right] \leq (b-a)\sqrt{\frac{log(2T)}{2m}} \tag{2.26}$$

**Demostración.** Se definen variables $X'_{ik} = -X_{ij}$ para $i = 1, ..., m$ y $k = 1, ..., T$ y se definen $S_{k+T} = \frac{1}{m}\sum_{i=1}^{m} X_{ik}$.

Puesto que cada $S'$ también es la media de m variables aleatorias independientes y centradas que toman valores en un intervalo de longitud $(b-a)$ se puede aplicar la Proposición previa para calcular:

$$\mathbb{E}\left[\sup_{1\leq j\leq T}|S_j|\right] = \mathbb{E}\left[\sup_{1\leq j\leq 2T} S_j\right] \leq (b-a)\sqrt{\frac{log(2T)}{2m}} \tag{2.27}$$

□

**Proposición 2.3.8** *Sea $\mathcal{H}$ una clase de reglas con función de crecimiento $\tau_{\mathcal{H}}$.*

*Entonces para toda distribución D:*

$$\mathbb{E}_{S\sim D^m}\left[\sup_{h\in\mathcal{H}}|L_S(h) - L_D(h)|\right] \leq \sqrt{\frac{2log(2\tau_{\mathcal{H}}(m))}{m}} \tag{2.28}$$

**Demostración.**

La esperanza del riesgo muestral es $L_D(h)$, por tanto, introduciendo la muestra $S' \sim D^m$ independiente de $S$ se tiene:

$$\begin{aligned}\mathbb{E}_{S\sim D^m}\left[\sup_{h\in\mathcal{H}}|L_S(h) - L_D(h)|\right] &= \mathbb{E}_{S\sim D^m}\left[\sup_{h\in\mathcal{H}}|\mathbb{E}_{S'\sim D^m}[L_S(h) - L_{S'}(h)]|\right] \\ &\leq \mathbb{E}_{S\sim D^m}\left[\sup_{h\in\mathcal{H}}\mathbb{E}_{S'\sim D^m}|L_S(h) - L_{S'}(h)|\right] \\ &\leq \mathbb{E}_{S,S'\sim D^m}\left[\sup_{h\in\mathcal{H}}|L_S(h) - L_{S'}(h)|\right]\end{aligned} \tag{2.29}$$

Sustituyendo los errores muestrales por su definición, donde denotamos por $(X_i, Y_i)$ con $i \leq m$ a los elementos de la muestra $S$ y por $(X'_i, Y'_i)$ con $i \leq m$ a los elementos de la muestra $S'$:

$$\mathbb{E}_{S\sim D^m}\left[\sup_{h\in\mathcal{H}}|L_S(h) - L_D(h)|\right] \leq \mathbb{E}_{S,S'\sim D^m}\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\left(\mathbb{I}_{h(X_i)\neq Y_i} - \mathbb{I}_{h(X'_i)\neq Y'_i}\right)\right|\right] \tag{2.30}$$

Se introducen ahora $\sigma_1, .., \sigma_m$ variables i.i.d. Rademacher independientes con respecto a las muestras $S$ y $S'$. Gracias a la Nota 2.3.2 se sabe que la parte derecha de (2.30) es igual a:

$$\mathbb{E}_{S,S'\sim D^m}\mathbb{E}_{\sigma\in U_\pm}\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\sigma_i\left(\mathbb{I}_{h(X_i)\neq Y_i} - \mathbb{I}_{h(X'_i)\neq Y'_i}\right)\right|\right] \tag{2.31}$$

Fijando $S = \{(X_1, Y_1), ..., (X_m, Y_m)\}$ y $S' = \{(X'_1, Y'_1), ..., (X'_m, Y'_m)\}$ se trata de acotar (2.32), teniendo en cuenta que cualquier cota válida para (2.32) es válida para (2.31):

$$\mathbb{E}_{\sigma \in U_\pm} \left[ \sup_{h \in \mathcal{H}} \left| \frac{1}{m} \sum_{i=1}^{m} \sigma_i \left( \mathbb{I}_{h(X_i) \neq Y_i} - \mathbb{I}_{h(X'_i) \neq Y'_i} \right) \right| \right] \tag{2.32}$$

Definiendo $C = \{X_1, ..., X_m, Y_1, ..., Y_m\}$ es claro que:

$$\left\{ \left| \frac{1}{m} \sum_{i=1}^{m} \sigma_i \left( \mathbb{I}_{h(X_i) \neq Y_i} - \mathbb{I}_{h(X'_i) \neq Y'_i} \right) \right| \right\}_{h \in \mathcal{H}} = \left\{ \left| \frac{1}{m} \sum_{i=1}^{m} \sigma_i \left( \mathbb{I}_{h(X_i) \neq Y_i} - \mathbb{I}_{h(X'_i) \neq Y'_i} \right) \right| \right\}_{h \in \mathcal{H}_C} \tag{2.33}$$

Donde $\mathcal{H}_C$ responde a la Definición 2.1.1.

Por lo que (2.32) es igual a:

$$\mathbb{E}_{\sigma \in U_\pm} \left[ \sup_{h \in \mathcal{H}_C} \left| \frac{1}{m} \sum_{i=1}^{m} \sigma_i \left( \mathbb{I}_{h(X_i) \neq Y_i} - \mathbb{I}_{h(X'_i) \neq Y'_i} \right) \right| \right] \tag{2.34}$$

Ahora bien, como $\sigma_i \left( \mathbb{I}_{h(X_i) \neq Y_i} - \mathbb{I}_{h(X'_i) \neq Y'_i} \right)$ son variables i.i.d. centradas que toman valores en $[-1, 1]$ aplicando el Corolario 2.3.7 se tiene que (2.34) es menor o igual que:

$$\sqrt{\frac{2log(2|\mathcal{H}_C|)}{m}} \tag{2.35}$$

Unificando todos los pasos se obtiene:

$$\mathbb{E}_{S \sim D^m} \left[ \sup_{h \in \mathcal{H}} |L_S(h) - L_D(h)| \right] \leq \sqrt{\frac{2log(2|\mathcal{H}_C|)}{m}} \leq \sqrt{\frac{2log(2\tau_{\mathcal{H}}(m))}{m}} \tag{2.36}$$

Como se buscaba demostrar.

□

**Corolario 2.3.9** *Sea $\mathcal{H}$ una clase de reglas con función de crecimiento $\tau_{\mathcal{H}}$. Entonces para toda distribución $D$ y para todo $\delta \in (0, 1)$, con probabilidad al menos $1 - \delta$ sobre la elección de la muestra se tiene:*

$$\sup_h |L_D(h) - L_S(h)| \leq \sqrt{\frac{2log(2\tau_{\mathcal{H}}(m))}{m\delta^2}} \tag{2.37}$$

**Demostración.** Puesto que $sup_{h \in \mathcal{H}} |L_D(h) - L_S(h)|$ es siempre positivo, combinando la desigualdad de Markov con la Proposición 2.3.8 se obtiene el resultado del Corolario. □

**Teorema 2.3.10** *Si la clase $\mathcal{H}$ tiene dimensión Vapnik-Chervonenkis igual a d, con $d < \infty$, entonces $\mathcal{H}$ tiene la propiedad de la convergencia uniforme.*

*Además:*

$$m_{\mathcal{H}}^{CU}(\epsilon, \delta) \leq \frac{8d}{(\delta\epsilon)^2} log\left(\frac{4d}{(\delta\epsilon)^2}\right) + \frac{4d}{(\delta\epsilon)^2} log(2^{1/d}e/d).$$

**Demostración.**

Dada $\mathcal{H}$ el Corolario 2.3.9 asegura que con probabilidad al menos $1 - \delta$:

$$\sup_h |L_D(h) - L_S(h)| \leq \sqrt{\frac{2log(2\tau_{\mathcal{H}}(m))}{m\delta^2}} \tag{2.38}$$

Además si se toma $m > d + 1$, se sabe gracias al Lema de Sauer 2.3.1:

$$\tau_{\mathcal{H}}(m) \leq (em/d)^d,$$

por lo que:

$$\sup_h |L_D(h) - L_S(h)| \leq \sqrt{\frac{2dlog(2^{1/d}em/d)}{m\delta^2}} \tag{2.39}$$

Para asegurar que $\sup_h |L_D(h) - L_S(h)| < \epsilon$ con probabilidad $1 - \delta$ se necesita tomar un m con:

$$m \geq \frac{2d}{(\delta\epsilon)^2} log(2^{1/d}me/d) = \frac{2d}{(\delta\epsilon)^2} log(m) + \frac{2d}{(\delta\epsilon)^2} log(2^{1/d}e/d). \tag{2.40}$$

Por el Lema 2.3.4 para que se satisfaga esta desigualdad es suficiente que:

$$m \geq \frac{8d}{(\delta\epsilon)^2} log\left(\frac{4d}{(\delta\epsilon)^2}\right) + \frac{4d}{(\delta\epsilon)^2} log(2^{1/d}e/d). \tag{2.41}$$

Dado que se tiene una cota superior de $m_{\mathcal{H}}^{CU}(\epsilon, \delta)$, $\mathcal{H}$ tiene la propiedad de la convergencia uniforme.

□

**Teorema 2.3.11** *(Teorema Fundamental del Aprendizaje Estadístico) Dada una clase de reglas $\mathcal{H}$ son equivalentes:*

1. *$\mathcal{H}$ tiene la propiedad de la convergencia uniforme.*
2. *$\mathcal{H}$ es PAC apendible.*
3. *$\mathcal{H}$ tiene dimensión-VC finita.*

**Demostración.**

1. $\Rightarrow$ 2. viene dado por la Proposición 1.3.1.

2. $\Rightarrow$ 3. viene dado por una consecuencia del Teorema de la Imposibilidad. Si la clase de reglas $\mathcal{H}$ es PAC aprendible, para cualesquiera $\epsilon, \delta \in (0,1)$, tomando una muestra de tamaño al menos $m_{\mathcal{H}}(\epsilon, \delta)$ y el algoritmo MRE, se cumplen las condiciones que el Corolario 2.2.3 exige para asegurar que la dimensión de Vapnik-Chervonenkis es finita.

3. $\Rightarrow$ 1. viene dado por el Teorema 2.3.10. $\square$

# Capítulo 3

# Otros paradigmas de aprendizaje

El hecho de que una clase de reglas $\mathcal{H}$ sea PAC aprendible es realmente fuerte. Permite encontrar una regla que puede 'competir' con el resto de reglas dada la exactitud ($\epsilon$) y seguridad ($\delta$) que se desee simplemente aumentando el tamaño de la muestra.

Las exigencias también son fuertes y se necesita que la dimensión Vapnik-Chervonenkis de $\mathcal{H}$ sea finita o equivalentemente que la clase $\mathcal{H}$ tenga la propiedad de la convergencia uniforme. Este hecho se deduce del Teorema 2.3.11, este Teorema se utiliza de ahora en adelante sin mención explícita al mismo.

Algunas clases de reglas interesantes no tienen dimensión-VC finita, se presenta a continuación el ejemplo de los clasificadores polinómicos. La regla $h_{p(x)}$ asociada al polinomio $p(x)$ asigna 1 a los $x$ que hacen a $p(x)$ positivo y 0 a los $x$ que hacen a $p(x)$ estrictamente negativo. La clase $\mathcal{H}_p$, formada por las reglas asociadas a polinómios de grado arbitrario no tiene dimensión Vapnik-Chervonenkis finita, puesto que dado un conjunto finito $C$ de puntos se puede encontrar un polinomio que tome los valores que se deseen en cada punto de $C$. Por tanto la clase $\mathcal{H}_p$ no es PAC aprendible.

Sin embargo la clase $\mathcal{H}_{p_n}$, formada por las reglas asociadas a polinómios de grado $n$ tiene dimensión Vapnik-Chervonenkis $n+1 < \infty$, puesto que un polinomio de grado $n$ solo puede cambiar de signo $n$ veces.

Estas clases son PAC aprendibles pero resultan pequeñas, puesto que en muchos problemas interesantes no se conocerá a priori el grado de la solución. Por ello nos interesa estudiar la clase:

$$\mathcal{H}_p = \underset{n\in\mathcal{N}}{\cup}\mathcal{H}_{p_n} \tag{3.1}$$

En busca de una definición adecuada del concepto de aprendizaje en este tipo de clases se define el paradigma del aprendizaje no uniforme. En este tipo de aprendizaje se pierde la uniformidad en el sentido de que se busca 'competir' contra una única regla de la clase dada, no contra todas las reglas en la clase como en el aprendizaje PAC.

**Definición 3.0.1** *(Apendizaje No Uniforme) Una clase de hipótesis $\mathcal{H}$ es aprendible no uniformemente si existe un algoritmo $A$ y una función $m_{\mathcal{H}}^{ANU} : (0,1)^2 \to \mathbb{N}$ tal que para todo $h \in \mathcal{H}$ y para todo $\epsilon, \delta \in (0,1)$, si $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, entonces para cualquier distribución $D$, con probabilidad al menos $1-\delta$ sobre la elección de la muestra $S \sim D^m$:*

$$L_D(A(S)) \leq L_D(h) + \epsilon.$$

Se llamará complejidad estadística de la clase $\mathcal{H}$ para el aprendizaje no uniforme a la mínima función $m_{\mathcal{H}}^{ANU}$ válida para la definición previa. El aprendizaje no uniforme es una relajación estricta del aprendizaje PAC, en el sentido de que toda regla aprendible PAC es aprendible no uniformemente.

## 3.1. Aprendizaje no uniforme

En esta sección se prueba que una clase de reglas $\mathcal{H}$ es aprendible no uniformemente sí y solo sí se puede escribir como unión numerable de clases $\mathcal{H}_n$ que tienen la propiedad de la convergencia uniforme.

En el caso del aprendizaje PAC el algoritmo MRE es válido para obtener las propiedades deseadas. Paralelamente, se prueba en esta sección que el algoritmo de minimización del riesgo estructural (MRS a partir de ahora), que se introduce en la Definición 3.1.4 es válido para conseguir las garantías deseadas para el aprendizaje no uniforme en cualquier clase de reglas aprendible no uniformemente.

Para definir de forma clara el algoritmo MRS se deben definir previamente algunos conceptos:

Sea $\mathcal{H}$ una clase de reglas que se puede escribir como $\underset{n\in\mathbb{N}}{\cup} \mathcal{H}_n$ donde las clases $\mathcal{H}_n$ tienen la propiedad de la convergencia uniforme:

**Definición 3.1.1** *Para cada $h \in \mathcal{H}$ se define:*

$$n(h) = \text{mín}\{n : h \in \mathcal{H}_n\}.$$

En el ejemplo previo sobre reglas asociadas a polinomios la regla $h_{x^3+2x+1}$ pertenece a todas las clases $\mathcal{H}_{p_n}$ con $n \geq 3$. Sin embargo la clase más simple a la que pertenece es $\mathcal{H}_{p_{n(h)}} = \mathcal{H}_{p_3}$.

**Definición 3.1.2** *Para cada $n \in \mathbb{N}$ se define $\epsilon_n : \mathbb{N} \times (0,1) \to (0,1)$ por:*

$$\epsilon_n(m, \delta) = min\{\epsilon \in (0,1) : m_{\mathcal{H}_n}^{CU}(\epsilon, \delta) \leq m\}.$$

**Nota 3.1.1** *Si $\mathcal{H}_n$ tiene la propiedad de la convergencia uniforme, para todo $m$ y para todo $\delta$ con probabilidad al menos $1-\delta$ sobre la elección de la muestra $S \sim D^m$:*

$$|L_D(h) - L_S(h)| \leq \epsilon_n(m, \delta) \tag{3.2}$$

*Para todo $h \in \mathcal{H}_n$.*

Si la clase $\mathcal{H}$ es una unión numerable de clases PAC aprendibles de distinta complejidad, téngase en mente el ejemplo de los polinomios, resulta conveniente cuantificar la confianza relativa que depositamos en cada clase. Esto se hace a traves de funciones peso, que se definen a continuación.

**Definición 3.1.3** *(Función peso) Se llama función peso a cualquier función $w : \mathbb{N} \to [0,1]$ con $\sum_{n=1}^{\infty} w(n) \leq 1$.*

Se describe ahora el funcionamiento del algoritmo MRS. En la práctica este algoritmo rara vez se puede llevar a cabo. Por ello una vez más se recalca que durante esta memoria se llama algoritmo a cualquier función que asigne una regla a cada muestra posible, sin tener en cuenta el coste computacional que tiene calcular dicha regla.

**Definición 3.1.4** *(Algoritmo MRS) Dada $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$ donde cada $\mathcal{H}_n$ tiene la propiedad de la convergencia uniforme, dado un $m \in \mathbb{N}$ y dada una función peso $w$, el algoritmo MRS devuelve un $h_{MRS}$ con:*

$$h_{MRS} \in \underset{h\in\mathcal{H}}{argmin} \left[ L_S(h) + \epsilon_{n(h)}(m, w(n(h))\delta) \right] \tag{3.3}$$

El algoritmo MRS se conoce en la literatura inglesa como SRM algorithm, respondiendo a las siglas de structural risk minimization.

**Proposición 3.1.2** *Dada un función peso $w$, y $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$ donde cada $\mathcal{H}_n$ tiene la propiedad de la convergencia uniforme, para todo $\delta \in (0,1)$ con probabilidad $1-\delta$ se tiene para todo $n \in \mathbb{N}$ y todo $h \in \mathcal{H}$:*

$$|L_D(h) - L_S(h)| \leq \epsilon_n(m, w(n)\delta) \tag{3.4}$$

Evidentemente fijada $h$ la mejor cota en (3.4) es $\underset{n:h\in\mathcal{H}_n}{\text{mín}} \epsilon_n(m, w(n)\delta)$ pero por simplicidad se usará $\epsilon_{n(h)}(m, w(n(h))\delta)$.

**Demostración.** Fijado un n, como $\mathcal{H}_n$ tiene la propiedad de la convergencia uniforme, del Lema 3.1.1 se deduce que con probabilidad al menos $1 - w(n)\delta$:

$$|L_D(h) - L_S(h)| \geq \epsilon_n(m, w(n)\delta) \text{ para todo } h \in \mathcal{H}_n \tag{3.5}$$

Por la desigualdad de Bonferroni la probabilidad de que la cota anterior se cumpla para todo $n$ es al menos:

$$1 - \sum_{n \in \mathbb{N}} w(n)\delta = 1 - \delta,$$

por lo que la Proposición queda probada. □

Probaremos a continuación que el algoritmo MRS es válido para garantizar el aprendizaje no uniforme en clases que se pueden escribir como unión numerable de clases PAC aprendibles.

**Teorema 3.1.3** *Si la clase $\mathcal{H}$ se puede escribir como unión numerable de clases $\mathcal{H}_n$ con la propiedad de la convergencia uniforme entonces $\mathcal{H}$ es aprendible no uniformemente.*

**Demostración.** Dada una función de peso $w$ y dado un $h \in \mathcal{H}$ se toma $m > m^{CU}_{\mathcal{H}_{n(h)}}(\epsilon/2, w(n(h))\delta)$ que verifica que $\epsilon/2 \leq \epsilon_{n(h)}(m, w(n)\sigma)$.

Sea $h_{MRS}$ la regla que nos devuelve el algoritmo MRS definido en la Definición 3.1.4. La Proposición 3.1.2 asegura que se cumplen (3.6) y (3.7) simultáneamente con probabilidad al menos $1 - \delta$:

$$\begin{aligned} L_D(h_{MRS}) &\leq L_S(h_{MRS}) + \epsilon_{n(h_{MRS})}(m, w(n)\sigma) \\ &= \min_{h' \in \mathcal{H}} \{L_S(h') + \epsilon_{n(h')}(m, w(n)\sigma)\} \\ &\leq L_S(h) + \epsilon_{n(h)}(m, w(n)\sigma) \end{aligned} \tag{3.6}$$

Y se tiene también que:

$$L_S(h) \leq L_D(h) + \epsilon_{n(h)}(m, w(n)\sigma) \tag{3.7}$$

Se deduce por tanto que con probabilidad al menos $1 - \delta$:

$$L_D(h) \leq L_D(h_{MRS}) + \epsilon_{n(h)}(m, w(n)\sigma) \leq L_D(h_{MRS}) + \epsilon \tag{3.8}$$

□

**Corolario 3.1.4** *La complejidad estadística para el aprendizaje no uniforme de una clase $\mathcal{H}$ que se puede escribir como unión numerable de clases $\mathcal{H}_n$ que tengan Dimensión de Vapnik-Chervonenkis finita está acotada por:*

$$m^{ANU}_{\mathcal{H}}(\epsilon, \delta, h) \leq m^{CU}_{\mathcal{H}_{n(h)}}(\epsilon/2, w(n(h))\delta) \tag{3.9}$$

**Demostración.** Se deduce directamente de la prueba del Teorema previo. □

Se ha probado que es suficiente que una clase $\mathcal{H}$ sea una unión numerable de clases $\mathcal{H}_n$ tienen dimensión Vapnik-Chervonenkis finita para que la clase $\mathcal{H}$ sea aprendible no uniformente. Se prueba a continuación que esta condición es tambien necesaria.

**Teorema 3.1.5** *La clase $\mathcal{H}$ es aprendible no uniformente sí y solo sí se puede escribir como unión numerable de clases $\mathcal{H}_n$ tienen dimensión Vapnik-Chervonenkis finita.*

**Demostración.**

En el Teorema 3.1.3 queda probada una implicación, se pueba a continuación que la implicación contraria también es cierta.

Sea $\mathcal{H}$ aprendible no uniformemente usando el algoritmo A. Sean $\epsilon, \delta \in (0,1)$ cumpliendo $\delta < \left(\frac{1}{2} - \epsilon\right) \frac{1}{1-\epsilon}$.

Se definen las clases $\mathcal{H}_n = \{h \in \mathcal{H} : m_{\mathcal{H}}^{ANU}(\epsilon, \delta, h) < n\}$. Esta definición junto con la definición de $m_{\mathcal{H}}^{ANU}$ implica que para cada $\mathcal{H}_n$ con probabilidad al menos $1-\delta$ sobre la elección de la muestra $S \sim D^m$ se tiene que $L_D(A(S)) \leq \epsilon$. Podemos aplicar por tanto el Corolario 2.2.3 para asegurar que $\mathcal{H}_n$ tiene dimensión Vapnik-Chervonenkis finita para todo $n \in \mathbb{N}$

□

Este resultado proporciona una caracterización para las funciones aprendibles no uniformemente. Se verá ahora que ciertas clases no tienen esta propiedad, como por ejemplo la clase de todas las funciones de $\mathbb{N}$ en $\{0,1\}$ no es aprendible no uniformemente. De hecho, toda clase que fragmenta a un conjunto infinito no es aprendible no uniformemente como prueba el siguiente Teorema.

**Teorema 3.1.6** *(Teorema de la imposibilidad en el aprendizaje no uniforme) Si $\mathcal{H}$ fragmenta a un conjunto $K$ infinito $\mathcal{H}$ entonces no es aprendible no uniformente.*

Las condiciones de este Teorema son más fuertes que las de la versión del Teorema para el aprendizaje PAC. En esa versión se exige que la clase fragmente a conjuntos de tamaño finito arbitrariamente grandes y ahora se exige que fragmente a un conjunto infinito.

**Demostración.**

Basta con probar que si $\mathcal{H}$ fragmenta a un conjunto $K$ infinito entonces para cualquier conjunto de clases $\{\mathcal{H}_n : n \in \mathbb{N}\}$ con $\mathcal{H} = \cup_{n\in\mathbb{N}}\mathcal{H}_n$ existe un $n$ tal que la dimensión de Vapnik-Chervonenkis de la clase $\mathcal{H}_n$ es infinita. Este resultado se prueba a continuación por reducción al absurdo, suponiendo que todos los $\mathcal{H}_n$ tienen dimensión-VC finita.

Se definen para todo $N \in \mathbb{N}$ los conjuntos $K_n$ que cumplen:

· $K_n \subset K$.

· $K_n$ no es fragmentable por $\mathcal{H}_n$.

· Si $m \neq n$ entonces $K_m \cap K_n = \emptyset$.

Es claro que dichos subconjuntos de $K$ existen, puesto que para cada $n$ basta coger un subconjunto de $VCdim(\mathcal{H}_n) + 1$ elementos de $K \setminus \cup_{i=1}^{n-1}\mathcal{H}_i$.

Por definición de la dimensión-VC para cada $n$ debe existir una función $f_n$ tal que para todo $h \in \mathcal{H}_n$ existe un $k \in K_n$ con $f_n(k) \neq h(k)$.

Se define la regla $f$ por:

$$f(k) = \begin{cases} f_n(k) & si \quad k \in K_n \\ \\ 0 & si \quad k \notin \cup K_n \end{cases} \tag{3.10}$$

Puesto que $\mathcal{H}$ fragmenta a $K$ debe existir $F \in \mathcal{H}$ con $F(k) = f(k)$ para todo $k \in K$.

Sin embargo para todo $n$, $F \notin \mathcal{H}_n$. Esto es absurdo pues $\mathcal{H} = \cup_{n\in\mathbb{N}}\mathcal{H}_n$.

□

## 3.2. La navaja de Ockham

El principio de la navaja de Ockham es bien conocido en economía y filosofía, este afirma que la explicación más sencilla suele ser la más probable.

Como se verá durante esta sección la noción de aprendizaje no uniforme nos permite dar soporte matemático a esta afirmación.

Para ello se considera una clase de reglas $\mathcal{H}$ que se puede escribir como unión numerable de clases formadas por una única regla, es decir, $\mathcal{H} = \cup_{n\in\mathbb{N}}\{h_n\}$. La clase de reglas $\mathcal{H}$ es aprendible no uniformemente por ser unión numerable de clases aprendibles PAC. Se formaliza a continuación el concepto de complejidad para reglas.

Dada una función $d : \mathcal{H} \to \{0,1\}^*$, donde $\{0,1\}^*$ es el conjunto de cadenas de ceros y unos finitas, se toma como medida de complejidad de una regla $h$ a $|d(h)|$.

Se estudia ahora como se puede utilizar esta definición de complejidad para construir una función peso. Para ello se aclara que se entiende por conjunto de cadenas libre de prefijos a cualquier $S \subset \{0,1\}^*$ tal que todo $s \in S$ no es un prefijo de ningún $s' \in S$ con $|s| < |s'|$, es decir, los $|s|$ primeros elementos de $s$ y $s'$ no son los mismos (teniendo en cuenta el orden).

**Proposición 3.2.1** *Sea la clase de reglas $\mathcal{H} = \cup_{n\in\mathbb{N}}\{h_n\}$ entonces para cada clase $\{h_n\}$:*

$$\epsilon_n(m,\delta) = \sqrt{\frac{log(2/\delta)}{2m}},$$

*donde $\epsilon_n$ está descito en la definición 3.1.2.*

*Además si se tiene la función $d : \mathcal{H} \to \{0,1\}^*$ donde $d(\mathcal{H})$ es un conjunto de cadenas libres de prefijos entonces*

$$w(n) = 2^{-|d(h_n)|}$$

*puede ser utilizada como función peso.*

**Demostración.**

De la desigualdad de Hoeffding 1.3.3 se deduce que:

$$\mathbb{P}[|L_S(h_n) - L_D(h_n)| > \epsilon] \leq 2exp(-2m\epsilon^2),$$

de donde:

$$\delta = 2exp\left(-2m\left(\epsilon_n(m,\delta)\right)^2\right)$$

$$\epsilon_n(m,\delta) = \sqrt{\frac{log(2/\delta)}{2m}}.$$

Para la segunda parte de la Proposición basta probar que $\sum_{n=1}^{\infty} w(h_n) \leq 1$. Considérese que se tira una moneda justa y se apunta 0 si sale cara y 1 si sale cruz. Además se deja de tirar la moneda cuando el número apuntado coincide con una cadena en $d(\mathcal{H})$.

Puesto que $d(\mathcal{H})$ es un conjunto de cadenas libre de prefijos todas ellas pueden coincidir con el número apuntado. En particular una cadena de longitud $|d(h)|$ coincidirá con probabilidad $2^{-|d(h)|}$, y puesto que la suma de las probabilidades no puede ser mayor que 1:

$$\sum_{n=1}^{\infty} w(n) = \sum_{n=1}^{\infty} 2^{-|d(h_n)|} \leq 1.$$

□

**Teorema 3.2.2** *Sea la clase de reglas $\mathcal{H} = \cup_{n\in\mathbb{N}}\{h_n\}$ y sea $d : \mathcal{H} \to S$ donde $S \subset \{0,1\}^*$ es un conjunto de cadenas libre de prefijos. Para todo $m$ y para todo $\delta \in (0,1)$ con probabilidad al menos $1-\delta$ sobre la elección de la muestra se tiene que para todo $h \in \mathcal{H}$:*

$$L_D(h) \leq L_S(h) + \sqrt{\frac{|d(h)| + log(2/\delta)}{2m}} \tag{3.11}$$

Obsérvese que la complejidad de la regla se penaliza, favoreciendo que las reglas más simples sean las de menor riesgo, como afirmó Ockham en el siglo XIV.

**Demostración.** Del Teorema 3.1.2, tomando $\epsilon_n(m,\delta) = \sqrt{\frac{log(2/\delta)}{2m}}$ y la función peso $w(n) = 2^{-|d(h_n)|}$ de la Proposición 3.2.1 se tiene el resultado de forma inmediata. □

Si en base al Teorema 3.2 se debiese escoger una regla se escogería aquella que minimizase:

$$L_S(h) + \sqrt{\frac{|d(h)| + log(2/\delta)}{2m}} \tag{3.12}$$

El algoritmo que escoge esta regla es conocido en el libro [3] como algoritmo de 'mínima longitud de descripción'.

## 3.3. Consistencia

Algunas clases de reglas no son aprendibles no uniformemente, en busca de un concepto de aprendizaje más vago se puede relaja la noción de aprendizaje no uniforme, dejando que la complejidad estadistica dependa no solo de $\epsilon$, $\delta$ y de $h \in \mathcal{H}$ sino también de la distribución subyacente $D$. Formalmente:

**Definición 3.3.1** *Sea $\mathcal{H}$ una clase de reglas y $\mathcal{P}$ un conjunto de distribuciones. El par $(\mathcal{H}, \mathcal{P})$ es aprendible consistentemente si existe un algoritmo $A$ y una función $m_{\mathcal{H}}^{CON} : (0,1)^2 \times \mathcal{H} \times \mathcal{P} \to \mathbb{N}$, tal que para todo $\epsilon, \delta \in (0,1)$, toda regla $h \in \mathcal{H}$ y toda distribución $D \in \mathcal{P}$, si $m > m_{\mathcal{H}}^{CON}(\epsilon, \delta, h, D)$, con probabilidad $1 - \delta$ sobre la elección de la muestra $S \sim D^m$ se tiene:*

$$L_D(A(S)) \leq L_D(h) + \epsilon \tag{3.13}$$

Se presenta a continuación un algoritmo válido para conseguir las garantías necesarias para la noción de aprendizaje consistente.

**Definición 3.3.2** *(Algoritmo Memorizar) En un problema de clasificación binaria, dada una muestra $S = \{(x_1, y_1), ..., (x_m, y_m)\}$, el algoritmo Memorizar devuelve la regla:*

$$h_{memo}(x) = \begin{cases} 1 & si \quad |\{i : (x_i, y_i) = (x, 1)\}| \geq |\{i : (x_i, y_i) = (x, 0)\}| \\ \\ 0 & si \quad |\{i : (x_i, y_i) = (x, 0)\}| \geq |\{i : (x_i, y_i) = (x, 1)\}| \end{cases}$$

En cierto modo este algoritmo es una estimación del predictor de Bayes, solo que este algoritmo solo conoce la muestra, no la distribución completa.

Quizás sorprende la aparición de un algoritmo que no generaliza, sino que como su propio nombre indica simplemente memoriza. En ciertas condiciones el algoritmo Memorizar es consistente, esto da una idea de lo vaga que es la noción de aprendizaje consistente. En particular, el algoritmo Memorizar es válido para probar que la clase de todos las reglas que etiquetan una cantidad numerable de atributos es aprendible consistentemente. A continuación se prueba esta propiedad del algoritmo Memorizar para el problema de clasificación binaria en el caso realizable, es decir, cuando fijada la distribución un mismo atributo no puede tener dos etiquetas distintas. En concreto en este caso el algoritmo Memorizar solo erra etiquetando atributos que no estan en la muestra.

Dada una distribución $\mathcal{D}$ se dota de un orden al conjunto numerable de atributos $\mathcal{H}$, de tal forma que $\mathcal{D}(x_i) \leq \mathcal{D}(x_j)$ sí y solo sí $i \leq j$ y con $\mathcal{D}(x) > 0$

para un conjunto infinito de atributos (si se cumple solo para un número finito de atributos la prueba es inmediata). Se tiene:

$$\lim_{n\to\infty}\sum_{i\geq n}\mathcal{D}(\{x_i\}) = \lim_{n\to\infty}\sum_{i=1}^{\infty}\mathcal{D}(\{x_i\}) - \lim_{n\to\infty}\sum_{i\leq n}\mathcal{D}(\{x_i\}) = 1-1=0.$$

Para todo $\epsilon > 0$ debe existir por tanto un $n_\epsilon$ con $\sum_{i\geq n_\epsilon}\mathcal{D}(\{x_i\}) < \epsilon$ y con $\mathcal{D}(\{x_{n_\epsilon-1}\}) < \mathcal{D}(\{x_{n_\epsilon}\})$. Denotando $\epsilon_\mathcal{D} = \mathcal{D}(\{x_{n_\epsilon}\}) > 0$ se tiene:

$$\mathcal{D}(\{x : \mathcal{D}(\{x_i\}) \leq \epsilon_\mathcal{D}\}) < \epsilon.$$

Además para $\eta > 0$, sea $n$ tal que $\mathcal{D}(\{x_i\}) < \eta$ para todo $i > n$, entonces para todo $m \in \mathbb{N}$:

$$\begin{aligned}\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\exists x_i : (\mathcal{D}(\{x_i\}) > \eta \ y \ x_i \notin S)] &\leq n \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[x \notin S : \mathcal{D}(\{x\}) > \eta] \\ &\leq n(1-\eta)^m \\ &= n(1-\frac{1}{\frac{1}{\eta}})^{\frac{1}{\eta}m\eta} \leq ne^{-m\eta}\end{aligned} \tag{3.14}$$

Se calcula ahora:

$$\begin{aligned}&\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_D(h_{memo}) < \epsilon] = \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\mathcal{D}(\{x : x \notin S\}) > \epsilon] \\ &\leq \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\mathcal{D}(\{x : \mathcal{D}(\{x\}) \leq \epsilon_\mathcal{D}\}) > \epsilon] + \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\mathcal{D}(\{x : \mathcal{D}(\{x\}) > \epsilon_\mathcal{D} \ y \ x \notin S\}) > \epsilon] \\ &\leq ne^{-m\epsilon_\mathcal{D}}\end{aligned} \tag{3.15}$$

Fijados $\epsilon, \delta > 0$ y tomando $m$ de modo que $ne^{-m\epsilon_\mathcal{D}} \leq \delta$ se obtiene que el algoritmo Memorizar es consistente con complejidad estadística $m_(\epsilon, \delta, \mathcal{D})$.

La clase de clasificadores binarios en un conjunto numerable es por tanto un ejemplo de clase consistente y no aprendible no uniformemente (Teorema 3.1.6).

# Capítulo 4

# Algunos algoritmos de clasificación

Como se ha estado insistiendo durante la memoria muchos de los algoritmos presentados generan problemas computacionales no tratables. El lector interesado tiene explicaciones más detalladas de esta afirmación en el Capítulo 8 de [3]. Esto ha motivado que se propongan otros procedimientos para obtener reglas de clasificación calculables. Son muy utilizados los basados en buscar fronteras lineales.

En este capítulo se consideran problemas en donde el espacio de atributos es $\mathbb{R}^d$ y se discuten algunos algoritmos de clasificación que pueden ser implementados de manera eficiente en la práctica. No se discutirá sobre los detalles de la implementación práctica de estos métodos, sino que al igual que en los capítulos anteriores se presentarán resultados sobre las propiedades estadísticas de estos métodos.

Se presentan tres grupos de algoritmos que tienen en común el hecho de estar basados en fronteras lineales. Se introducen en primer lugar los clasificadores lineales discutiendo la conveniencia de sustituir la función de pérdida 0-1 por otras más tratables. Luego se presentan las máquinas de soporte vectorial, que están basadas también en principio en la busqueda de fronteras lineales, pero que a través de una transformación manejada de forma implícita permiten enriquecer el tipo de fronteras admisibles. Y se termina con una incursión en las redes neuronales, uno de los métodos más utilizados en la actualidad que también está originado en la clasificación lineal.

## 4.1. Clasificadores lineales

Durante esta sección se presentan las clases de reglas basadas en clasificadores lineales. Estas reglas vienen determinadas por el par $(\mathbf{w}, b) \in \mathbb{R}^d \times \mathbb{R}$ y se definen como:

$$h_{(\mathbf{w},b)}(x) = signo(\langle \mathbf{w}, \mathbf{x} \rangle + b), \tag{4.1}$$

donde $signo(r) : \mathbb{R} \to \{-1, 1\}$ está dada por:

$$signo(0) = 1 \text{ y para } r \neq 0, \ signo(r) = \frac{r}{|r|}.$$

Se advierte que en esta subsección las etiquetas son $\{1, -1\}$. El cambio a la codificación $\{1, 0\}$ es obvio.

Es claro que el semiespacio $\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle + b \geq 0\}$ es el conjunto de los puntos que $h_{(\mathbf{w},b)}$ etiqueta con 1. A veces se dice que el semiespacio etiqueta a los atributos. Además, por simplificar el lenguaje, se habla indistintamente del par $(\mathbf{w}, b)$ y del semiespacio generado por él.

**Nota 4.1.1** *Definiendo* $\mathbf{w}' = (b, w_1, ..., w_n)$ *y* $\mathbf{x}' = (1, x_1, ...x_n)$ *es claro que* $\mathbf{x}$ *pertenece al semiespacio de* $\mathbb{R}^n$ *determinado por* $\mathbf{w}$ *y* $b$ *sí y solo sí* $\mathbf{x}'$ *pertenece al semiespacio homogéneo de* $\mathbb{R}^{n+1}$ *determinado por* $\mathbf{w}'$ *pues:*

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle.$$

Por ese motivo, a partir de ahora en vez de trabajar con el semiespacio dado se trabajará en una dimensión más con el semiespacio homogéneo equivalente. Y análogamente al caso no homogéneo se habla indistintamente de $(\mathbf{w})$ y del semiespacio homogéneo generado por él.

Se demuestra a continuación que la dimensión Vapnik-Chervonenkis de las clases de semiespacios es finita. Esto prueba que en estas clases el algoritmo MRE es válido para conseguir las garantías necesarias para el aprendizaje PAC y además será útil a la hora de calcular la complejidad estadística de las redes neuronales.

**Teorema 4.1.2** *La dimensión Vapnik-Chervonenkis de la clase de semiespacios de* $\mathbb{R}^d$ *es* $d + 1$ *y la de la clase de semiespacios homogéneos de* $\mathbb{R}^{d+1}$ *es* $d + 1$.

**Demostración.** Por la Nota 4.1.1 ambas afirmaciones son equivalentes, por tanto se probará únicamente que la dimensión Vapnik-Chervonenkis de la clase de semiespacios homogéneos de $\mathbb{R}^{d+1}$ es $d + 1$.

Es claro que la clase de semiespacios homogéneos de $\mathbb{R}^{d+1}$ fragmenta a la base canónica de $\mathbb{R}^{d+1}$, $(e_1, ..., e_{d+1})$. Pues tomando $w = (y_1, ..., y_{d+1})$ para todo $i$ $\langle \mathbf{w}, e_i \rangle = y_i$.

Solo falta probar que la clase de semiespacios homogéneos no fragmenta a ningún conjunto de $d + 2$ vectores.

Sean $\mathbf{x}_1, ..\mathbf{x}_{d+2} \in \mathbb{R}^{d+1}$. Supóngase que el conjunto de estos vectores es fragmentado por la clase de semiespacios homogéneos.

Deben existir $a_1, ..., a_{d+2}$ con $\sum_{i=1}^{d+2} a_i \mathbf{x}_i = \mathbf{0}$.

Sean $I = \{i : a_i > 0\}$ y $J = \{i : a_i < 0\}$, entonces:

$$\sum_{i \in I} a_i \mathbf{x}_i = \sum_{i \in J} |a_i| \mathbf{x}_i \tag{4.2}$$

Puesto que la clase de semiespacios homogéneos fragmenta al conjunto debe existir un $\mathbf{w}$ con $\langle \mathbf{w}, \mathbf{x}_i \rangle > 0$ para todo $i$, por lo que:

$$0 < \sum_{i \in I} a_i \langle \mathbf{x}_i, \mathbf{w} \rangle = \langle \sum_{i \in I} a_i \mathbf{x}_i, \mathbf{w} \rangle = \langle \sum_{i \in J} |a_i| \mathbf{x}_i, \mathbf{w} \rangle = \sum_{i \in J} |a_i| \langle \mathbf{x}_i, \mathbf{w} \rangle < 0 \tag{4.3}$$

Si uno de los dos conjuntos ($I$ o $J$) es vacío entonces una de las dos desigualdades se convertiría en una igualdad pero la contradicción no desaparece. Esto concluye la prueba. □

Este resultado evidencia que no siempre existe un semiespacio que etiquete la muestra correctamente. En caso de que dicho semiespacio exista se dice que la muestra es separable.

Tomando la función de pérdida 0-1, si la muestra es separable el algoritmo MRE clasifica correctamente todos los elementos de la muestra, es decir, la muestra es separable sí y solo sí la regla MRE satisface:

$$\frac{1}{m} \sum_{i=1}^{n} l_{0-1}(\mathbf{w}_{MRE}, (\mathbf{x}, y)) = 0.$$

En esta situación el algoritmo Perceptron calcula una solución de la siguiente forma:

·Dada una muestra $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$

·Se inicializa en $\mathbf{w}_0 = \mathbf{0}$

·Y para $k = 0, 1, ...$ e $i$ con $l_{0-1}(\mathbf{w}_k, (\mathbf{x}_i, y_i)) \neq 0$:

Se actualiza $\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{x}_i y_i$

En el capítulo 9 de [3] se prueba que el algoritmo Perceptron calcula una solución en a lo sumo $(RB)^2$ pasos, donde $R = \text{máx}\, ||x_i||$ y $B$ es el margen de separación (que se define en la sección 4,2).

Si por el contrario la muestra no es separable, el cáculo de la regla MRE se convierte en un problema np-duro por lo que conviene buscar un algoritmo alternativo.

La fuente de problemas en la clasificación lineal, con pérdida 0-1 está en la falta de buenas propiedades de esta función de peso. Esto se entiende mejor por comparación con el problema de regresión lineal. En este problema se trata de encontrar la función lineal del atributo $\mathbf{x}$ que mejor aproxima a la etiqueta $y \in \mathbb{R}$ en el sentido de los mínimos cuadrados, es decir, se maneja la función de pérdida $l_2(\mathbf{w}, (\mathbf{x}, y)) = |y - \langle \mathbf{x}, \mathbf{w} \rangle|^2$ de manera que el riesgo empírico es:

$$\frac{1}{m}\sum_{i=1}^{n}|y-\langle \mathbf{x},\mathbf{w}\rangle|^2,$$

y el cálculo del MRE se reduce a resolver el problema de mínimos cuadrados, que es equivalente a la resolución de un sistema lineal.

De vuelta al problema de clasificación la pérdida cuadrática no parece la más adecuada. Aunque es difícil conseguir que una función de pérdida nos lleve a un problema computacionalmente tan manejable como el de mínimos cuadrados hay una clase de funciones de pérdida que son manejables de manera razonable desde el punto de vista computacional: las funciones de pérdida convexas. El problema de MRE correspondiente es un problema de optimización convexa que se puede resolver numéricamente mediante un método de descenso del gradiente o alguna de sus variantes, como se trata en el capítulo 14 de [3].

Una posible función de pérdida convexa para la clase de clasificadores lineales es la función de pérdida logística:

$$l_{log}(\mathbf{w},(\mathbf{x},y)) = log(1+exp(-y\langle \mathbf{w},\mathbf{x}\rangle)) \tag{4.4}$$

El método de regresión logística busca el $\mathbf{w}$ que minimiza:

$$L_{log}(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{n} log(1+exp(-y\langle \mathbf{w},\mathbf{x}\rangle)) \tag{4.5}$$

Esta función, en oposición a la función que describía el riesgo para la función de pérdida 0-1, es convexa en $\mathbf{w}$ y diferenciable. Se puede usar por tanto un método de descenso de gradiente para encontrar un minimizador.

La función de pérdida logística considera que cuanto más alejado esté el atributo de la frontera con mayor seguridad este ha sido clasificado correctamente. Esta función de pérdida da lugar al problema de regresión logística.

Se presenta en la sección siguiente una forma distinta de definir una función de pérdida sobre los clasificadores lineales de tal forma que la minimización del riesgo empírico pueda ser llevada a cabo.

## 4.2. Las máquinas de soporte vectorial

En esta sección se presenta una herramienta realmente útil en el aprendizaje estadístico: las máquinas de soporte vectorial (MSV a partir de ahora). En la literatura inglesa esta herramienta es conocida como SVM respondiendo a las siglas de support vector machine. Se distingue entre dos algoritmos, un algoritmo MSV-duro para el caso separable y un algoritmo MSV-suave para cuando la muestra no es separable. El poder de esta herramienta reside en que su complejidad estadística no depende de la dimensión del espacio de atributos sino únicamente de la muestra. Esto se entenderá mejor al leer la subsección 4.2.1.

Además aunque el MSV es en principio un método de clasificación lineal, el algoritmo depende únicamente de los productos escalares entre los elementos de la muestra, es decir, de la matriz de Gram. Esto permite manejar transformaciones de los atributos (clasificaremos a partir de las imágenes de transformaciones $\phi(\mathbf{x})$ en lugar de a partir de $\mathbf{x}$) simplemente conociendo los valores $\langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$. La transformación $\phi$ permite transportar los datos a un espacio de dimensión superior e incluso a espacios de dimensión infinita, esto aumenta la separabilidad, puesto que la clase de semiespacios homogéneos con la que estamos tratando tiene dimensión-VC igual a la dimensión del espacio (Teorema 4.1.2).

La regla obtenida es lineal en el espacio transformado pero puede ser no lineal en el espacio original, esto aporta mayor flexibilidad a las reglas MSV. Además la transformación $\phi$ se puede manejar de forma implícita, porque solo se necesita conocer el núcleo $K(x, y) = \langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle$. Este hecho (al que se suele alucir como el 'kernel trick') aumenta la flexibilidad en el uso de métodos MSV.

En el capítulo previo se menciona que si una muestra es separable entonces el algoritmo Perceptron encuentra un semiespacio que clasifique correctamente la muestra. Sin embargo, normalmente habrá infinitos semiespacios correctos. El algoritmo MSV-duro busca la regla asociada a un semiespacio que mejor separa la muestra, en el sentido de que los puntos de la muestra estén lo más alejados posibles del hiperplano separador.

Se entiende intuitivamente que este semiespacio es mejor puesto que es el más estable, en el sentido de que es el que mayores variaciones permite en los elementos de la muestra de tal forma que su etiqueta no cambie.

Para formalizar el problema se observa en primer lugar que la distancia entre un punto $\mathbf{x}$ y el hiperplano dado por $\mathbf{w}$ con $||\mathbf{w}|| = 1$ es $|\langle\mathbf{w}, \mathbf{x}\rangle|$. Para comprobarlo basta tener en cuenta que:

La distancia entre un punto $\mathbf{x}$ y el hiperplano $\mathbf{w}$ es:

$$min\{||\mathbf{x} - \mathbf{v}'|| : \langle\mathbf{w}, \mathbf{v}'\rangle = 0\}.$$

Se toma $\mathbf{v} = \mathbf{x} + (\langle\mathbf{w}, \mathbf{x}\rangle)\mathbf{w}$ y se prueba que en este punto se alcanza el mínimo previo. Puesto que $||\mathbf{w}|| = 1$, $\mathbf{v}$ satisface:

$$\langle\mathbf{w}, \mathbf{v}\rangle = \langle\mathbf{w}, \mathbf{x}\rangle - (\langle\mathbf{w}, \mathbf{x}\rangle)||\mathbf{w}||^2 = 0,$$

y además, para cualquier otro punto $\mathbf{u}$ cumpliendo $\langle\mathbf{w}, \mathbf{u}\rangle = 0$. Puesto que $\langle\mathbf{w}, \mathbf{v} - \mathbf{u}\rangle = \langle\mathbf{w}, \mathbf{v}\rangle - \langle\mathbf{w}, \mathbf{u}\rangle = 0$ se tiene:

$$\begin{aligned} ||\mathbf{x} - \mathbf{u}||^2 &= ||\mathbf{x} - \mathbf{v} + \mathbf{v} - \mathbf{u}||^2 \\ &= ||\mathbf{x} - \mathbf{v}||^2 + ||\mathbf{v} - \mathbf{u}||^2 + 2\langle\mathbf{x} - \mathbf{v}, \mathbf{v} - \mathbf{u}\rangle \\ &\geq ||\mathbf{x} - \mathbf{v}||^2 + 2\langle\mathbf{x} - \mathbf{v}, \mathbf{v} - \mathbf{u}\rangle \\ &= ||\mathbf{x} - \mathbf{v}||^2 + 2(\langle\mathbf{w}, \mathbf{x}\rangle)\langle\mathbf{w}, \mathbf{v} - \mathbf{u}\rangle \\ &= ||\mathbf{x} - \mathbf{v}||^2, \end{aligned} \tag{4.6}$$

por lo que:

$$\begin{aligned} min\{||\mathbf{x}-\mathbf{v}'|| : \langle \mathbf{w}, \mathbf{v}' \rangle = 0\} &= ||\mathbf{x}-\mathbf{v}|| \\ &= |\langle \mathbf{w}, \mathbf{x} \rangle| \; ||\mathbf{w}|| \\ &= |\langle \mathbf{w}, \mathbf{x} \rangle|. \end{aligned} \tag{4.7}$$

Si $(\mathbf{x}_i, y_i)$ es un elemento de la muestra, obsérvese que si el semiespacio asociado a $\mathbf{w}$ clasifica correctamente $\mathbf{x}_i$ entonces $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle)$ es positivo. Esto ayuda a formalizar el concepto de margen:

**Definición 4.2.1** *(Margen) Dada una muestra* $S = \{(\mathbf{x}_1, y_i), ..., (\mathbf{x}_m, y_m)\}$ *y un semiespacio dado por* $\mathbf{w}$ *con* $||\mathbf{w}|| = 1$ *que separa correctamente la muestra, es decir,* $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle) > 0$ *para todo* $i \in [m]$*, el margen del semiespacio es:*

$$B = \min_{i \in [m]} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle).$$

El margen está definido solo para semiplanos que separan correctamente la muestra y es la menor distancia entre un punto de la muestra y el hiperplano separante.

A continuación se define formalmente el algoritmo MSV-duro:

**Definición 4.2.2** *(MSV-duro) Dada una muestra separable* $S = \{(\mathbf{x}_1, y_i), ..., (\mathbf{x}_m, y_m)\}$ *el algoritmo MSV-duro devuelve el semiespacio:*

$$\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{||\mathbf{w}_0||},$$

*donde:*

$$\mathbf{w}_0 \in \underset{(\mathbf{w},b)}{argmin} ||\mathbf{w}||^2 \; s.a \; y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle) \geq 1 \tag{4.8}$$

El problema de MSV-duro es un problema de optimización convexa, pues la norma cuadrática es una función convexa y las restricciones son lineales. Cada restricción es un semiespacio, por ello los valores de $\mathbf{w}$ válidos están en la intersección de $n$ semiespacios y esta intersección será necesariamente un conjunto convexo, un politopo.

**Proposición 4.2.1** *Si la muestra es separable el semiespacio que devuelve el algoritmo MSV-duro maximiza el margen.*

**Demostración.** Sea $\hat{\mathbf{w}}$ el semiespacio devuelto por el algoritmo MSV-duro obsérvese que se puede recuperar $\mathbf{w}_0$ pues:

$$|\mathbf{w}_0| \geq \frac{1}{\underset{i\in[m]}{min}\ y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle)},$$

y como $\mathbf{w}_0$ es solución de (4.8) se debe cumplir la igualdad.

Si existiese otro semiespacio $\mathbf{w}'$ que separa correctamente la muestra con $||\mathbf{w}'|| = 1$ tal que:

$$\underset{i\in[m]}{min}\ y_i(\langle \mathbf{w}', \mathbf{x}_i\rangle) > \underset{i\in[m]}{min}\ y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle),$$

definiendo:

$$\mathbf{w'}_0 = \frac{\mathbf{w}'}{\underset{i\in[m]}{min}\ y_i(\langle \mathbf{w}', \mathbf{x}_i\rangle)},$$

es claro que $\mathbf{w'}_0$ cumple las condiciones en (4.8) y que $||\mathbf{w'}_0|| < ||\mathbf{w}_0||$ contradiciendo el hecho de que el algoritmo MSV-duro devolviese $\hat{\mathbf{w}}$. □

En caso de que la muestra no sea separable, algo bastante común en la práctica, el algoritmo MSV-duro no devuelve ningún semiespacio, por ello se deben relajar las exigencias de dicho algoritmo.

Se plantea el algoritmo MSV-suave que permite una holgura en las restricciones del algoritmo previo para que existan soluciones, pero que al mismo tiempo penaliza esta holgura para que esta sea pequeña.

**Definición 4.2.3** *Dado un $\lambda > 0$ y dada una muestra separable $S = \{(\mathbf{x}_1, y_i), ..., (\mathbf{x}_m, y_m)\}$ el algoritmo MSV-suave devuelve el semiespacio:*

$$\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{||\mathbf{w}_0||},$$

*donde:*

$$\mathbf{w}_0 \in \underset{\mathbf{w}}{argmin} \left( \lambda||\mathbf{w}||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \right), \tag{4.9}$$

*con $y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle) \geq 1 - \xi_i$ donde $\xi_i > 0$.*

Obsérvese que el parámetro $\lambda$ mide la penalización que sufren las holguras permitidas, a menor $\lambda$ mayor penalización.

Recordando la definición de la función de pérdida hinge (ecuación (1.3)) y utilizando $\phi(\mathbf{w}(\mathbf{x}, y)) = 1 - y(\langle \mathbf{w}, \mathbf{x}\rangle)$ se tiene:

$$l_{hinge}(\mathbf{w}, (\mathbf{x}, y)) = \text{máx}\{0, 1 - y(\langle \mathbf{w}, \mathbf{x}\rangle)\}.$$

Esto permite reescribir (4.9) como:

$$\mathbf{w}_0 \in \underset{\mathbf{w}}{argmin} \left( L_S^{hinge}(\mathbf{w}) + \lambda ||\mathbf{w}||^2 \right), \tag{4.10}$$

donde:

$$L_S^{hinge}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} \text{máx}\{0, 1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle)\}.$$

La ecuación (4.10) proporciona una interpretación alternativa del MSV, ahora este se puede ver como un método MRE penalizado para la función de pérdida hinge, donde la penalización es proporcional a la norma del vector normal.

### 4.2.1. Complejidad estadística del algoritmo MSV-suave

Para calcular la complejidad estadística asociada al algoritmo MSV-suave se usa el Corolario 13.8 de [3]. Este afirma que si se utiliza una función de pérdida convexa y $\rho$-Lipschitz, cualquier $w^*$ con:

$$w^* \in \underset{\mathbf{w}}{argmin} L_S(\mathbf{w}) + \lambda ||w||^2 \tag{4.11}$$

satisface que:

$$\underset{S \sim D^m}{\mathbb{E}} [L_D^{hinge} w^*] \leq \underset{w}{\text{ínf}} \left[ L_D^{hinge}(\mathbf{w}) + \lambda ||\mathbf{w}||^2 \right] + \frac{2\rho^2}{\lambda m} \tag{4.12}$$

Obsérvese que la regla $w^*$ en (4.11) para la función de pérdida hinge es la regla que devuelve el algoritmo MSV-suave.

**Lema 4.2.2** *Fijado* $(\mathbf{x}, y)$ *la función de pérdida hinge es convexa y* $\rho$*-Lipschitz con* $||x|| \leq \rho$.

**Demostración.** La función de pérdida hinge se puede escribir como $f \circ g$ con $f(z) = \text{máx}\{0, z\}$ y $g(w) = -y\langle \mathbf{w}, \mathbf{x} \rangle$ donde ambas funciones son convexas y además $f$ es creciente de forma clara.

La composición $f \circ g$ de dos funciones convexas, donde $f$ es creciente, es convexa:

$$\begin{aligned} f(g(\alpha w + (1-\alpha)x)) &\leq f(g(\alpha w) + g((1-\alpha)x)) \\ &\leq f(g(\alpha w)) + f(g((1-\alpha)x)) \end{aligned} \tag{4.13}$$

Por lo tanto la función de pérdida hinge es convexa, véase que también es $\rho$-Lipschitz con $||x|| \leq \rho$:

$$\begin{aligned} |l_{hinge}(\mathbf{w}_1,(\mathbf{x},y)) - l_{hinge}(\mathbf{w}_2,(\mathbf{x},y))| &= |f(-y\langle \mathbf{w}_1,\mathbf{x}\rangle) - f(-y\langle \mathbf{w}_2,\mathbf{x}\rangle)| \\ &\leq |-y\langle \mathbf{w}_1,\mathbf{x}\rangle + y\langle \mathbf{w}_2,\mathbf{x}\rangle| \\ &\leq |\langle \mathbf{w}_1,\mathbf{x}\rangle + \langle \mathbf{w}_2,\mathbf{x}\rangle| \\ &\leq ||\mathbf{w}_1 - \mathbf{w}_2||\ ||\mathbf{x}|| \\ &\leq \rho||\mathbf{w}_1 - \mathbf{w}_2|| \end{aligned} \tag{4.14}$$

Donde en la primera desigualdad hemos utilizado que $f$ es 1-Lipschitz. □

Si los atributos toman valores en la bola $B(0,\rho)$ la función de pérdida hinge es convexa y $\rho$-Lipschitz, por ello del Corolario 13.8 de [3] se deduce que:

$$\mathop{\mathbb{E}}_{S\sim D^m}[L_D^{hinge}(\mathbf{w}(S))] \leq \inf_{\mathbf{w}}\left[L_D^{hinge}(\mathbf{w}) + \lambda||\mathbf{w}||^2\right] + \frac{2\rho^2}{\lambda m} \tag{4.15}$$

Donde $\mathbf{w}(S)$ es la regla que devuelve el algoritmo MSV-suave dada la muestra $S$.

**Nota 4.2.3** *La función de pérdida 0-1 se puede acotar por la función de pérdida hinge, puesto que:*

*·Si w clasifica erróneamente a* $\mathbf{x}$ *entonces* $y(\langle \mathbf{w},\mathbf{x}\rangle) < 0$ *y por tanto* $l_{0,1}(\mathbf{w},(\mathbf{x},y)) = 1 \leq 1 - y(\langle \mathbf{w},\mathbf{x}\rangle) = l_{hinge}(\mathbf{w},(\mathbf{x},y))$.

*· Y si w clasifica correctamente a* $\mathbf{x}$ *entonces* $y(\langle \mathbf{w},\mathbf{x}\rangle) > 0$ *y por tanto* $l_{0,1}(\mathbf{w},(\mathbf{x},y)) = 0 \leq \text{máx}\{0, 1 - y(\langle \mathbf{w},\mathbf{x}\rangle)\} = l_{hinge}(\mathbf{w},(\mathbf{x},y))$.

Esto nos permite acotar el riesgo asociado a la pérdida 0-1:

$$\mathop{\mathbb{E}}_{S\sim D^m}[L_D^{0-1}(\mathbf{w}(S))] \leq L_D^{hinge}(\mathbf{w}) + \lambda||\mathbf{w}||^2 + \frac{2\rho^2}{\lambda m} \textit{ para todo } \mathbf{w} \tag{4.16}$$

Además para todo $B > 0$ se puede tomar $\lambda = \sqrt{\frac{2\rho^2}{B^2 m}}$ y considerando solo los $\mathbf{w}$ con $||\mathbf{w}|| \leq B$ se tiene:

$$\begin{aligned} \mathop{\mathbb{E}}_{S\sim D^m}[L_D^{0-1}(\mathbf{w}(S))] &\leq \min_{w:||\mathbf{w}||\leq B} L_D^{hinge}(\mathbf{w}) + \sqrt{\frac{2\rho^2}{B^2 m}}B^2 + \frac{2\rho^2}{\sqrt{\frac{2\rho^2}{B^2 m}}m} \\ &= \min_{w:||\mathbf{w}||\leq B} L_D^{hinge}(\mathbf{w}) + \sqrt{\frac{2B^2\rho^2}{m}} + \sqrt{\frac{2B^2\rho^2}{m}} \\ &= \min_{w:||\mathbf{w}||\leq B} L_D^{hinge}(\mathbf{w}) + \sqrt{\frac{8B^2\rho^2}{m}} \end{aligned} \tag{4.17}$$

Se ha comentado al principio de esta subsección que el MSV tiene la flexibilidad de producir fronteras no lineales a traves de el uso de una transformación implícita

(se suele aludir a esto como 'kernel trick') . Se discuten a continuación los detalles. El $\mathbf{w}$ óptimo para el MSV-suave se puede escribir como:

$$\hat{\mathbf{w}} = \sum_{i=1}^{n} y_i \hat{\alpha} \mathbf{x}_i,$$

donde $\hat{\alpha} = (\hat{\alpha}_i)_{i=1}^n$, es solución del problema:

$$\text{máx} \left( -\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right),$$

sujeto a

$$\sum_{i=1}^{n} y_i \alpha_i = 0, \ \sum_{i=1}^{n} \alpha_i = 1 \ y \ 0 \leq \alpha_i \leq 1/\lambda.$$

Véase por ejemplo el capítulo 7 de [4], en donde se prueba esta equivalencia. Este nuevo problema depende de los atributos únicamente a través de la matriz de Gram. Esto tiene la siguiente implicación. Supongamos que $\phi$ es una aplicación de $\mathcal{X}$ en un espacio de Hilbert. Entonces se puede tratar de usar la regla MSV-suave a partir de los datos transformados, $\phi(\mathbf{x}_i)$. Por la observación anterior bastaría conocer los valores $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ para poder resolver el problema equivalente. Entonces el hiperplano que definiría la regla óptima correspondiente sería el $f(\mathbf{x}) = 0$ con $f(\mathbf{x}) = \langle \phi(\mathbf{x}), (\hat{w}) \rangle$. Por linealidad se obtiene que

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \hat{\alpha}_i K(\mathbf{x}_i, \mathbf{x}).$$

Entonces observamos que por un lado se pueden emplear los atributos transformados para la clasificación sin manejar la transformación de manera explícita, porque solo se necesita conocer el núcleo $K(x, y) = \langle \phi(x), \phi(y) \rangle$. Por otro lado de vuelta al espacio original la función $f$ obtenida por este procedimiento ya no es necesariamente lineal y esto es lo que le da esa mayor flexibilidad al método MSV con núcleo. Conviene de todas formas observar que la garantía probabilística que se ha dado en (4.16) depende de la acotación a priori sobre la norma de los atributos. El uso de la transformación $\phi$ puede tener mucha influencia sobre la cota, cosa que rara vez es tenida en cuenta en la práctica.

## 4.3. Redes Neuronales

Basándose en el modelo cerebral que valió un Nobel a Ramón y Cajal se empezó a desarrollar a mediados del siglo XX el paradigma de aprendizaje de las redes neuronales.

A día de hoy las redes neuronales se han convertido en uno de los métodos más utilizados. Frecuentemente son la base de los procedimientos utilizados para ganar los desafíos de construcción de inteligencias artificiales, como por ejemplo el desafío de clasificación de dígitos manuscritos basado en la base de datos MNIST (ver http://yann.lecun.com/exdb/mnist). El éxito de las redes neuronales se debe a la gran cantidad de funciones que pueden representar.

### 4.3.1. Descripción de una Red Neuronal

Las redes neuronales que se estudian durante esta memoria quedan descritas por:

· Un grafo dirigido (V,E) en el que llamaremos neuronas a los nodos. Además V se debe poder escribir como unión disjunta de conjuntos $V_0, ..., V_T$ de tal forma que cada arista de E que sale desde una neurona de $V_t$ va hacia una neurona de $V_{t+1}$. Se suele denominar capas a los conjuntos $V_0, ..., V_n$, además se llamará capa de atributos a $V_0$, capa de etiquetas a $V_n$ y capas ocultas al resto de los subconjuntos. Las neuronas de la capa $V_t$ se denotan por $v_{t,i}$ con $i \leq |V_t|$.

· Una función de peso sobre las aristas $w : E \to \mathbb{R}$. Se denota por $w(v_{t,i}, v_{t+1,j})$ al peso de la arista que va de la neurona $v_{t,i}$ a la neurona $v_{t+1,j}$. Se debe destacar que aunque se denominen igual, no se corresponde con la función de peso definida en el capítulo 3.

· Una función escalar $\sigma_v : \mathbb{R} \to \mathbb{R}$ asociada a cada neurona denominada función de 'activación'. Aunque se puede elegir una función de 'activación' distinta para cada neurona, se suele tomar la misma función para todas las neuronas. Aquí se usará la función *signo* definida al comienzo de la sección 4.1, otras funciones de 'activación' usadas en la práctica son la función *sigmoide* y la función *hinge*.

Se representa en la Figura 4.1 una red neuronal con tres capas:

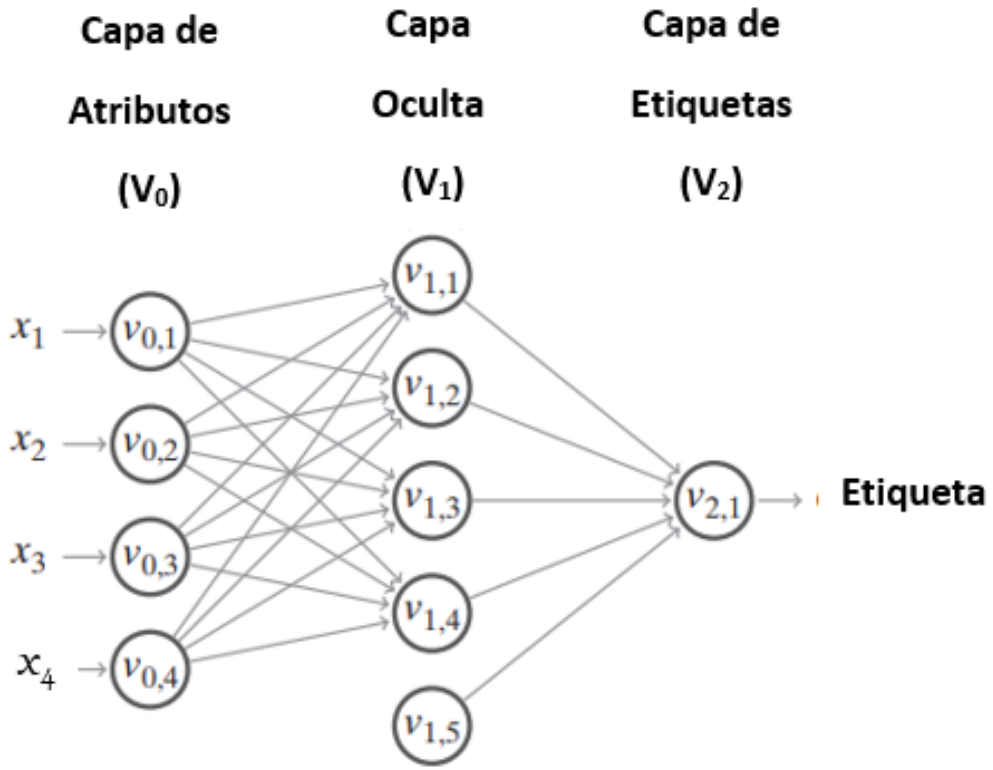


Figura 4.1: Grafo de una red neuronal. Imagen extraída de [3]

Partiendo de los valores de las neuronas de la capa de atributos, el valor de la neurona $v_{t+1,i}$, denotado por $o_{t+1,i}$, se calcula a partir de los valores de las neuronas de la capa previa de acuerdo con la siguiente fórmula recursiva:

$$o_{t+1,i} = signo\left(\sum_{r:(v_{t,r},v_{t+1,i})} o_{t,r} w(v_{t,r}, v_{t+1,i})\right) \tag{4.18}$$

Se entiende por estructura de red neuronal al grafo (V,E) y a las funciones de activación. La clase de reglas dada por una estructura de red neuronal es el conjunto de reglas que se pueden obtener modificando las funciones peso. Esta clase se denota por $\mathcal{H}_{V,E,\sigma}$

### 4.3.2. Complejidad estadística

Como en el caso de las MSV es interesante conocer la complejidad estadística de la clase de reglas dada por una estructura de red neuronal.

Será realmente útil a la hora de calcular la complejidad estadística deseada recordar las siguientes definiciones:

• Dada una clase de reglas H y un conjunto $C = \{c_1, ..., c_m\}$ se define la *restricción de* $\mathcal{H}$ *a* $C$ como el conjunto de regla de C a {0,1}

$$\mathcal{H}_C = \{(h(c_1), ..., h(c_m) : h \in \mathcal{H}\}.$$

• Sea $\mathcal{H}$ una clase de reglas. Se define la *función de crecimiento* de $\mathcal{H}$, denotada por $\tau_H(m) : \mathbb{N} \to \mathbb{N}$, como:

$$\tau_H(m) = \sup_{C \subset X : |C| = m} |H_C|.$$

Además se necesitarán los siguientes resultados auxiliares:

**Lema 4.3.1** *Sea* $\mathcal{F}_i$ *un conjunto de funciones de* $\mathcal{X}$ *a* $\mathcal{Y}$ *para* $i = 1, 2$. *Sea* $\mathcal{H} = \mathcal{F}_1 \times \mathcal{F}_2$ *la clase del producto cartesiano entonces:*

$$\tau_{\mathcal{H}} \leq \tau_{\mathcal{F}_1} \tau_{\mathcal{F}_2}.$$

**Demostración.** Fijado un m:

$$\begin{aligned}
\tau_{\mathcal{F}_1}(m)\tau_{\mathcal{F}_2}(m) &= \max_{C:|C|=m} |\mathcal{F}_{1C}| \max_{C':|C'|=m} |\mathcal{F}_{2C}| \\
&= \max_{C:|C|=m} |\{(h(c_1), ..., h(c_m) : h \in \mathcal{F}_1\}| \max_{C':|C'|=m} |\{(g(c'_1), ..., g(c'_m) : g \in \mathcal{F}_2\}| \\
&\geq \max_{C:|C|=m} |\{(h(c_1), ..., h(c_m) : h \in \mathcal{F}_1\}||\{(g(c_1), ..., g(c_m) : g \in \mathcal{F}_2\}| \\
&= \max_{C:|C|=m} |\{(h(c_1) \times g(c_1), ..., h(c_m) \times g(c_m) : h \times g \in \mathcal{F}_1 \times \mathcal{F}_2\}| \\
&= \max_{C:|C|=m} |\mathcal{H}_C| = \tau_{\mathcal{H}}(m)
\end{aligned} \tag{4.19}$$

□

**Lema 4.3.2** *Sea* $\mathcal{F}_1$ *un conjunto de funciones de* $\mathcal{Y}$ *a* $\mathcal{Z}$ *y sea* $\mathcal{F}_2$ *un conjunto de funciones de* $\mathcal{X}$ *a* $\mathcal{Y}$ *para* $i = 1, 2$. *Sea* $\mathcal{H} = \mathcal{F}_1 \circ \mathcal{F}_2$ *la clase de la composición de las clases entonces:*

$$\tau_{\mathcal{H}} \leq \tau_{\mathcal{F}_1} \tau_{\mathcal{F}_2}.$$

**Demostración.** Fijado un m:

$$\begin{aligned}
\tau_{\mathcal{F}_1}\tau_{\mathcal{F}_2} &= \max_{C:|C|=m} |\mathcal{F}_{1C}| \max_{C':|C'|=m} |\mathcal{F}_{2C'}| \\
&= \max_{C:|C|=m} |\{(h(c_1), ..., h(c_m) : h \in \mathcal{F}_1\}| \max_{C':|C'|=m} |\{(g(c'_1), ..., g(c'_m) : g \in \mathcal{F}_2\}| \\
&\geq \max_{C:|C|=m} |\{(h(c_1), ..., h(c_m) : h \in \mathcal{F}_1\}||\{(g(h(c_1)), ..., g(h(c_m)) : g \in \mathcal{F}_2\}| \\
&\geq \max_{C:|C|=m} |\{(g(h(c_1)), ..., g(h(c_m)) : g \circ h \in \mathcal{F}_2 \circ \mathcal{F}_2\}| \\
&= \max_{C:|C|=2m} |\mathcal{H}_C| = \tau_{\mathcal{H}}
\end{aligned} \tag{4.20}$$

□

Con estas herramientas se puede demostrar el resultado principal de esta sección:

**Teorema 4.3.3** *La función de crecimiento de la clase de reglas dada por una estructura de red neuronal está acotada por:*

$$\tau_{\mathcal{H}_{V,E,\sigma}}(m) \leq (em)^{|E|}.$$

**Demostración.** Se denota $\mathcal{H} = \mathcal{H}_{V,E,\sigma}$ para simplificar la notación. Puesto que es una clase de reglas de clasificación binaria tiene sentido hablar de la función de crecimiento $\tau_{\mathcal{H}}(m)$.

Asignando valores a las aristas que van de la capa $V_{t-1}$ a la capa $V_t$ se obtienen funciones que van de $\mathbb{R}^{|V_{t-1}|}$ a $\{-1,1\}^{|V_t|}$. Definiendo $\mathcal{H}_t$ como la clase de todas las funciones posibles de $\mathbb{R}^{|V_{t-1}|}$ a $\{-1,1\}^{|V_t|}$, $\mathcal{H}$ puede escribirse como composición de estas clases: $\mathcal{H} = \mathcal{H}_T \circ ... \circ \mathcal{H}_1$. Gracias al Lema 4.3.2 se tiene que la función de crecimiento de $\mathcal{H}$ está acotada por:

$$\tau_{\mathcal{H}}(m) \leq \prod_{t=1}^{T} \tau_{\mathcal{H}_t}(m) \tag{4.21}$$

Además cada clase $\mathcal{H}_t$ puede escribirse como el producto cartesiano: $\mathcal{H}_t = \mathcal{H}_{t,1} \times ... \times \mathcal{H}_{t,|V_t|}$ donde $\mathcal{H}_{t,i}$ es la clase de funciones posibles de $\mathbb{R}^{|V_{t-1}|}$ a $\{-1,1\}$. Por tanto gracias al Lema 4.3.1:

$$\tau_{\mathcal{H}_t}(m) \leq \prod_{i=1}^{|V_t|} \tau_{\mathcal{H}_{t,i}}(m) \tag{4.22}$$

Dado que cada neurona es una regla lineal asociada a un semiespacio homogéneo, por el Teorema 4.1.2, la dimensión de semiespacios homogéneos es la dimensión del espacio de atributos, por tanto denotando por $d_{t,i}$ al número de aristas que llegan a $v_{t,i}$ y utilizando el Lema de Sauer 2.3.1:

$$\tau_{\mathcal{H}_{t,i}}(m) \leq \left(\frac{em}{d_{t,i}}\right)^{d_{t,i}} \tag{4.23}$$

Combinando (4.21), (4.22) y (4.23) se obtiene:

$$\tau_{\mathcal{H}}(m) \leq (em)^{\sum_{t,i} d_{t,i}} = (em)^{E} \tag{4.24}$$

□

**Corolario 4.3.4** *La dimensión Vapnik-Chervonenkis de la clase de reglas dada por una estructura de red neuronal está acotada por:*

$$VCdim(\mathcal{H}) \leq O(|E|log(|E|)).$$

**Demostración.** Para calcular la dimensión-VC se asume que $\mathcal{H}$ fragmenta a un conjunto de m puntos, entonces por el Teorema previo:

$$2^m \leq (em)^{|E|}$$

Finalmente aplicando el Lema 2.3.4:

$$m \leq |E| log(em)/log(2),$$

y es trivial que esta cota debe ser cumplida también por la dimensión-VC de $\mathcal{H}$. □

Combinando el resultado del Teorema 2.3.10 que acota la complejidad estadística mediante una función de la dimensión Vapnik-Chervonenkis con el hecho de que la dimensión-VC esté acotada por una función creciente de $|E|$ se deduce que la complejidad estadística está acotada por una función creciente de $|E|$. Es decir, el número de aristas proporciona una cota para el tamaño necesario de la muestra para aprender sobre esta clase. En particular a mayor número de aristas un mayor tamaño de muestra es necesario.

Se comenta al inicio de la sección que las redes neuronales pueden representar una gran cantidad de funciones, para dar garantías al lector sobre la certeza de esta afirmación se comenta el hecho de que una red neuronal de solo una capa oculta es capáz de expresar cualquier función de $\{0,1\}^n \to \{0,1\}$ como se prueba en la sección 20.3 de [3]. Aunque es cierto que la capa oculta debe tener un tamaño exponencial con respecto a n. Imagínese por tanto, con el desarrollo del Deep Learning (que consiste en entrenar redes neuronales con una cantidad enorme de capas ocultas) la gran 'expresividad' que se puede alcanzar con el uso de las redes neuronales.

# Bibliografía


[1] Buhlmann, P. y van de Geer, S., (2011). '*Statistics for High-Dimensional Data Methods*', Theory and Applications. Springer.

[2] R. M. Dudley,(2014), '*Uniform Central Limit Theorems*', Cambridge University Press.

[3] S. Shalev-Shwartz y S. Ben-David, (2014), '*Understanding Machine Learning*', Cambridge University Press.

[4] John Shawe-Taylor y Nello Cristianini,(2004), '*Kernel Methods for Pattern Analysis*', Cambridge University Press.

[5] Valiant, L. G., (1984). '*A theory of the learnable*', Communications of the ACM 27, 1134–1142.

[6] Vladimir N. Vapnik,(2004), '*The Nature of Statistical Learning Theory*', Springer-Verlag New York, Inc.