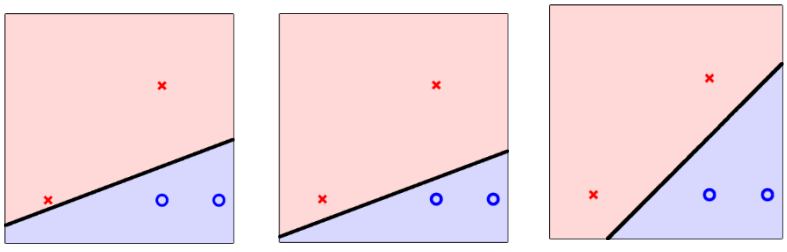
e-Capítulo 8

Máquinas de vectores soporte

Los modelos lineales son potentes. La transformada no lineal y la red neuronal (una cascada de modelos lineales) son herramientas que aumentan su poder expresivo. Aumentar la potencia expresiva tiene un precio: el sobreajuste y el tiempo de cálculo. ¿Podemos obtener la potencia expresiva sin pagar el precio? La respuesta es sí. Nuestro nuevo modelo, la máquina de vectores soporte (SVM), utiliza un "colchón de seguridad" al separar los datos. Como resultado, la SVM es más robusta al ruido, lo que ayuda a combatir el sobreajuste. Además, la SVM puede trabajar sin problemas con una nueva y potente herramienta llamada *kernel:* una forma computacionalmente eficiente de utilizar transformaciones no lineales de alta dimensión (¡incluso infinita!). Cuando combinamos el colchón de seguridad de la SVM con el "truco del núcleo", el resultado es un potente y eficiente modelo no lineal con regularización automática. El modelo SVM es popular porque funciona bien en la práctica y es fácil de usar. Este capítulo presenta las matemáticas y los algoritmos que hacen funcionar la SVM.

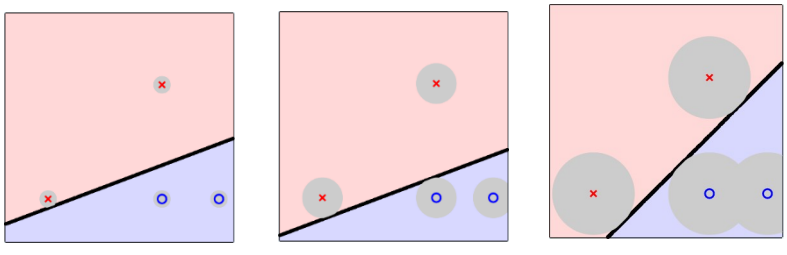
8.1 El hiperplano óptimo

Volvamos al modelo de perceptrón del capítulo 1. Las ilustraciones siguientes sobre un conjunto de datos de juguete le ayudarán a refrescar la memoria. En 2 dimensiones, un perceptrón intenta separar los datos con una línea, lo que es posible en este ejemplo.



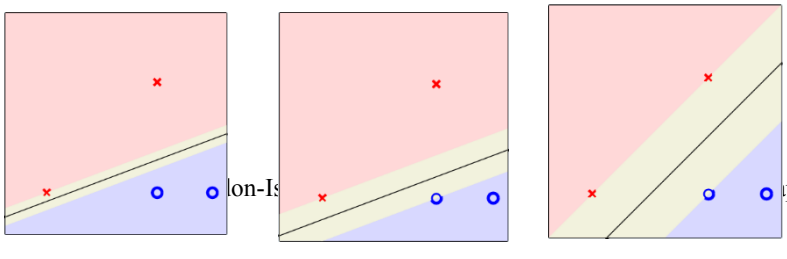
Como puede ver, muchas líneas separan los datos y el Algoritmo de Aprendizaje Perceptrón (PLA) encuentra una de ellas. ¿Nos importa cuál encuentra el PLA? Todos los separadores tienen A,q = 0, por lo que el análisis VC del capítulo 2 da el mismo límite Aqqt para cada separador. Bueno, el límite VC puede decir una cosa, pero seguramente nuestra intuición dice que se prefiere el separador más a la derecha .

Intentemos concretar un argumento que apoye nuestra intuición. En la práctica, existen errores de medición, es decir, ruido. Coloca regiones sombreadas idénticas alrededor de cada punto de datos, siendo el radio de la región la cantidad de posible error de medición. El verdadero punto de datos puede encontrarse en cualquier lugar dentro de esta "región de incertidumbre" debido al error de medición. Un separador es "seguro" con respecto al error de medición si clasifica correctamente los puntos de datos falsos. Es decir, no importa en qué parte de su región de incertidumbre se encuentre el punto de datos verdadero, sigue estando en el lado correcto del separador. La siguiente figura muestra los mayores errores de medición seguros para cada separador.



Un separador que puede tolerar más errores de medición es más seguro. El separador más a la derecha tolera el mayor error, mientras que para el separador más a la izquierda, incluso un pequeño error en algunos puntos de datos podría dar lugar a una clasificación errónea. En el Capítulo 4, vimos que el ruido (por ejemplo, el error de medición) es la principal causa del sobreajuste. La regularización nos ayuda a combatir el ruido y evitar el sobreajuste. En nuestro ejemplo, el separador situado más a la derecha es más robusto al ruido sin comprometer la entrada; está mejor "regularizado". Nuestra intuición está bien justificada.

También podemos cuantificar la tolerancia al ruido desde el punto de vista del separador. Colocamos un cojín a cada lado del separador. Llamamos a este separador con un cojín /n/, y decimos que separa los datos si ningún punto de datos se encuentra dentro de su cojín. He aquí el mayor cojín que podemos colocar alrededor de cada uno de nuestros tres separadores candidatos.



Para obtener el cojín más grueso, sigue extendiendo el cojín por igual a ambos lados del separador hasta que llegues a un punto de datos. El grosor refleja la cantidad de ruido que puede tolerar el separador. Si cualquier punto de datos se ve perturbado como máximo por el grosor de cualquiera de los lados del cojín, seguirá estando en el lado correcto del separador. El grosor máximo (tolerancia al ruido) posible de un separador se denomina morpin. Nuestra intuición nos dice que elijamos el separador más gordo posible, el que tenga el máximo margen. En esta sección abordaremos tres cuestiones importantes:

1. ¿Podemos encontrar eficazmente el separador más gordo?
2. ¿Por qué es mejor un separador gordo que uno fino?
3. ¿Qué hacer si los datos no son separables?

La primera cuestión se refiere al algoritmo; la segunda, a €..t i y, la tercera, a €ip (también profundizaremos en esta cuestión en la sección 8.4).

Nuestra discusión se refería a 2 dimensiones. En dimensiones superiores, el separador es un hiperplano y nuestra intuición sigue siendo válida. He aquí un calentamiento.

Ejercicio 8.1

Supongamos que @ contiene dos puntos de datos (x+, II) y (xc , -1) . Demuestre que

(a) Ningún hiperplano puede tolerar un radio de ruido sup$\frac{1}{2}$‖$\mathbf{x}_+ - \mathbf{x}_-$‖.

(b) Existe un hiperplano que tolera un radio de ruido $\frac{1}{2}\|\mathbf{x}_+ - \mathbf{x}_-\|$.

8.1.1 Encontrar el hiperplano de separación más grueso

Antes de pasar a las matemáticas, fijamos una convención que utilizaremos a lo largo del capítulo. Recordemos que un hiperplano se define por sus pesos w.

Aislamiento del sesgo. Dividimos explícitamente el vector de pesos w de la siguiente manera. El peso de sesgo m9 se elimina, y el resto de los pesos *w , . md* permanecen en

w. La razón es que las matemáticas tratarán estos dos tipos de pesos de forma diferente. Para evitar confusiones, cambiaremos el nombre de m9 por b (por sesgo), pero seguiremos utilizando w para *(w , d)*

Capítulos anteriores	Este capítulo
$\mathbf{x} \in \{1\} \times \mathbb{R}^d;\ \mathbf{w} \in \mathbb{R}^{d+1}$	$\mathbf{x} \in \mathbb{R}^d;\ b \in \mathbb{R},\ \mathbf{w} \in \mathbb{R}^d$
$h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x})$	$_d \qquad w_d$
1 w_0	
x w =	

b=
sesgo

x w

d

w_d

h(x)-
sign(w'xTd)

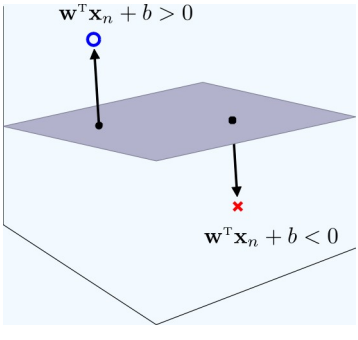
Un hiperplano de separación de margen máximo tiene dos propiedades definitorias.

1. Separa los datos.
2. Tiene el cojín más grueso entre los hiperplanos que separan los datos.

Para encontrar un hiperplano de *separación* con el máximo margen, primero reexaminamos la definición de hiperplano de separación y la transformamos en una definición equivalente y más conveniente. A continuación, discutiremos cómo calcular el margen de cualquier hiperplano de separación dado (de modo que podamos encontrar el que tenga el máximo margen). Como observamos en nuestra discusión intuitiva anterior, el margen se obtiene extendiendo el cojín hasta que se llega a un punto de datos. Es decir, el margen es la distancia desde el hiperplano hasta la puntuación/punto de datos. Así pues, tenemos que familiarizarnos con la geometría de los hiperplanos; en particular, con la forma de calcular la distancia de un punto de datos al hiperplano.

Hiperplanos de separación. El hiperplano fi, definido por (fi, w), separa los datos si y sólo si para n = 1, ... , *N,*

$$\#q(w*x + b) > 0. \qquad (8.1)$$



La señal pp (w*xq + b) es positiva para cada punto de datos. Sin embargo, la magnitud de la señal no es significativa por sí misma, ya que podemos hacerla arbitrariamente pequeña o grande /o /fie algún *hiperplano* revelando los pesos y el sesgo. Esto se debe a que (b, w) es el *mismo* hiperplano que (b/p, w/p) para cualquier p
0. Reescalando los pesos, podemos controlar el tamaño de la señal para nuestros puntos de datos. Escojamos un valor concreto de *p,*

$$\rho = \min_{\cdot\cdot} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b),$$

que es positivo debido a (8.1). Ahora, reescalamos los pesos para obtener el hiperplano (b/p, w/p). Para estos pesos reescalados

$$\min \quad pp \begin{pmatrix} w * \\ - xq + - \end{pmatrix} \quad \begin{matrix} b \\ \end{matrix} \quad \begin{matrix} 1 \\ \cdot \end{matrix} \quad \min \quad y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = \frac{\rho}{\cdot} = 1.$$

Así, para cualquier hiperplano de separación, siempre es posible elegir pesos de modo que todas las señales pq(w*xq -I- b) sean de magnitud mayor o igual a 1, con igualdad satisfecha por al menos una (xp, pq). Esto motiva nuestra nueva definición de hiperplano de separación.

Definición 8.1 (Hiperplano de separación) . El hiperplano fi separa los datos si

y sólo si puede ser representado por pesos (b, w) que satisfagan

$$\min_p (w^* x_p - I - b) = 1. \qquad (82)$$

Las condiciones (8.1) y (8.2) son equivalentes. *Todo hiperplano de* separación puede acomodarse a la definición 8.1. Todo lo que hicimos fue restringir la forma en que n/pehrnicnf/p representamos tal hiperplano eligiendo una normalización (dependiente de los datos) para los pesos, para asegurar que la magnitud de la señal sea significativa. Nuestra normalización en (8.2) será particularmente conveniente para de- rivar el algoritmo para encontrar el separador de margen máximo. El siguiente ejercicio ofrece un ejemplo concreto de la renormalización de los pesos para satisfacer (8.2).

Ejercicio 8. 2

Considere los datos siguientes y un "hiperplano" (6, w) que separa los datos.

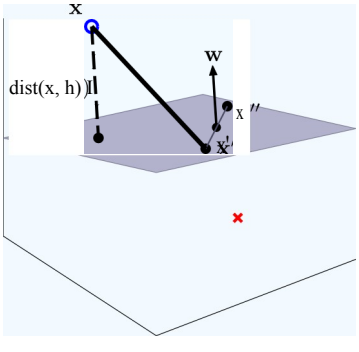
$$X = \begin{matrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \end{matrix} \qquad y = \begin{matrix} -1 \\ -1 \\ +1 \end{matrix} \qquad \mathbf{w} = \begin{vmatrix} 1.2 \\ -3.2 \end{vmatrix} \; h = -0.5$$

(a) Compute $\rho = \min_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)$.

(b) Calcule los pesos - (b, w) y demuestre que satisfacen (8.2}.

(c) Traza ambos hiperplanos para demostrar que son el *mismo* separador.

Margen de un hiperplano. Para calcular el margen de un hiperplano de separación, necesitamos calcular la distancia desde el hiperplano al punto de datos más pequeño. Para empezar, calculemos la distancia desde un punto arbitrario x a un hiperplano de separación fi - (h, w) que satisfaga (8.2). Anotemos esta distancia por dist(x, fi). Refiriéndonos a la figura de la derecha, dist(x, fi) es la longitud de la perpendicular de x a fi. Sea x' un punto cualquiera del hiperplano, lo que significa que w*x' -I- h = 0. Sea u un vector unitario normal al hiperplano fi.



Entonces, dist(x, fi) = | u * (x - x') | , la proyección del vector (x - x') sobre u. Ahora argumentamos que w es normal al hiperplano, y por tanto podemos tomar u w/ | w| . De hecho, cualquier vector situado en el hiperplano puede expresarse mediante (x" - x') para algunos x', x" en el hiperplano, como se muestra. Entonces, usando w*x -fi para puntos en el hiperplano,

$$A'(X'' - XQ - G'X'' - G'X' - -ñ+6- 0.$$

Por lo tanto, w es ortogonal a cada vector en el hiperplano, por lo tanto es el vector normal como se afirma. Estableciendo u w/ | w| , la distancia de x a fi es

dist(x, fi) - | u"(x - x') | - $\frac{\mathbf{x} - \mathbf{w}^{\mathrm{T}}\mathbf{x}'|}{\|\mathbf{w}\|}$ $\frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|}{\|\mathbf{w}\|}$,

donde usamos w*x' = -b en el último paso, ya que x' está en fi. Ahora puede ver por qué separamos el sesgo fi de las ponderaciones w: el cálculo de la distancia trata estos dos parámetros de forma diferente.

Ahora estamos preparados para calcular el margen de un hiperplano de separación. Consideremos los puntos de datos x , ... , xp, y el hiperplano fi (b, w) que satisface la condición de separación (8.2). Puesto que pp - -1-1,

$$|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b| = |y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)| = y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b),$$

donde la última igualdad se deduce porque (h, w) separa los datos, lo que implica que 9q(w*xq + b) es positivo. Por lo tanto, la distancia de un punto de datos a fi es

$$\mathsf{dist}(\mathbf{x}_n, h) = \frac{y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}{\|\mathbf{w}\|}.$$

Por lo tanto, el punto de datos más cercano al hiperplano tiene una distancia

$$\text{mindist(xp , fi)} \frac{\text{-1}}{\|\mathbf{w}\|} \cdot \min_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = \frac{1}{\|\mathbf{w}\|},$$

donde la última igualdad se deduce porque (b, w) separa los datos y satisface (8.2). Esta sencilla expresión para la distancia del punto de datos más cercano al hiperplano es la única razón por la que elegimos normalizar (h, w) como lo hicimos, exigiendo (8.2).' Para cualquier hiperplano de separación que satisfaga (8.2), el margen es 1/ ||w ||. Si aguantas un poco más, estás a punto de cosechar todo el beneficio, a saber, un algoritmo sencillo para encontrar el hiperplano óptimo (más gordo).

El hiperplano de separación de máximo margen. El hiperplano de separación de margen máximo (h*, w*) es el que satisface la condición de separación (8.2) con una norma de peso mínima (ya que el margen es la inversa de la norma de peso). En lugar de minimizar la norma-peso, podemos equivalentemente minimizar yw*w, que es analíticamente más amigable. Por lo tanto, para encontrar este hiperplano óptimo, tenemos que resolver el siguiente problema de optimización.

$$\begin{aligned} &\underset{\text{b, w}}{\text{minimizar:}} && \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{sujeto a:} && \min \quad \text{pq(w*xq + b) = 1.} \end{aligned} \tag{8.3}$$

La restricción garantiza que el hiperplano separa los datos según (8.2). Obsérvese que el sesgo b no aparece en la cantidad que se está minimizando, pero está implicado en la restricción (de nuevo, fi se trata de forma diferente a w). Para que el problema de optimización sea más fácil de resolver, podemos sustituir la única restricción

' Algo de terminología: los parámetros (b, w) de un hiperplano de separación que satisfacen (8.2) se denominan representación canónica del hiperplano. Para un hiperplano de separación en su representación canónica, el margen es simplemente la norma inversa de los pesos.

mente 9q (w*xp + b) = 1 con *N* restricciones 'más laxas' pp (w*xq -I-) para n = 1, ... , *N* y resolver el problema de optimización:

$$\underset{b,\mathbf{w}}{\text{minimizar:}} \quad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \qquad (84)$$

sujeto a: pp (w*xp -t- fi) 1 (n - 1 ,,
N).

La restricción de (8.3) implica las restricciones de (8.4), lo que significa que las restricciones de (8.4) son más laxas. Afortunadamente, n/ tfie solución *óptima*, las restricciones en (8.4) se convierten en equivalentes a la restricción en (8.3), siempre y cuando haya ejemplos positivos y negativos en los datos. Después de resolver (8.4), demostraremos que la restricción de (8.3) se satisface automáticamente. Esto significa que también habremos resuelto (8.3).

Para ello, utilizaremos una prueba por contradicción. Supongamos que la solución (h*, w*) de (8.4) tiene

$$\rho^* = \min_n y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n + b^*) > 1,$$

y por lo tanto no es una solución a (8.3). Consideremos el hiperplano reescalado $(b, \mathbf{w})$ (b*, $\frac{1}{\rho}$w*), que satisface las restricciones de (8.4) por construcción. Para (b, w), tenemos que ||w||| ||w* |||w* || (a menos que w* 0), lo que significa que w* no puede ser óptimo para (8.4) a menos que w* - 0. No es posible tener w* = 0 ya que esto no clasificaría correctamente los ejemplos positivos y negativos en los datos.

Nos referiremos a este hiperplano de separación más gordo como /fie *óptimo hiper-*
plano. Para obtener el hiperplano óptimo, basta con resolver el problema de optimización de (8.4).

Ejemplo 8.2. La mejor manera de entender lo que está pasando es trabajar cuidadosamente a través de un ejemplo para ver cómo resolver el problema de optimización en (8.4) resulta en el hiperplano óptimo (h*, w*). En dos dimensiones, un hiperplano se especifica mediante los parámetros (b, i 2). La matriz de datos y los valores objetivo, junto con las restricciones de separabilidad de (8.4) se resumen a continuación. La desigualdad en una fila particular es la restricción de separabilidad para el punto de datos correspondiente en esa fila.

$$\mathrm{X} \quad \begin{matrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{matrix} \quad \mathrm{y} \quad \begin{matrix} -\mathrm{l} \\ -\mathrm{l} \\ . \\ +\mathrm{l} \end{matrix} \quad \begin{matrix} -\mathrm{b} > 1 & \text{(i)} \\ -(2\mathrm{vj} - \mathrm{l} - 2 < \mathrm{z}) 1 & \text{(Îi)} \\ 2\mathrm{vt} - \mathrm{t} - \mathrm{h} 1 & \text{(iii)} \\ 3\,\mathrm{mt} + \mathrm{b} 1 & \text{(iv)} \end{matrix}$$

Combinando (i) y (iii) se obtiene

$$w_1 \geq 1.$$

Combinando (ii) y (iii) se obtiene

$$w_2 \leq -1.$$

Esto significa que $\frac{1}{2}$(m$ + m2) 1 con igualdad cuando mt1 y m2 = -1.

Se puede comprobar fácilmente que

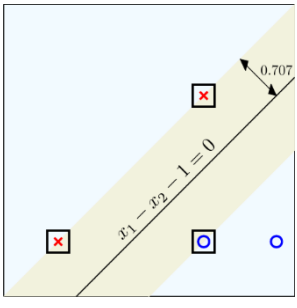
$$(b^* = -1, w_1^* = 1, w_2^* = -1)$$

satisface las cuatro restricciones, minimiza$\frac{1}{2}$ (v2 -I- v2), y por lo tanto da el hiperplano óptimo. El hiperplano óptimo se muestra en la siguiente figura.

Hvperplano óptimo

#(x) = sign(i -i - 1)

margin: $\frac{1}{\|\mathbf{w}^*\|} = \frac{1}{\sqrt{2}} \approx 0.707.$



Los puntos de datos (i), (ii) y (iii) se recuadran porque se cumplen sus restricciones de separación: pp (w**xp -I- b*) = 1.

Para los puntos de datos que cumplen exactamente sus restricciones, dist(xp , p) = $\frac{1}{\|\mathbf{w}^*\|}$. -Estos puntos de datos se sitúan en el límite del cojín y desempeñan un papel importante. Se denominan rectores de apoyo. En cierto sentido, los vectores de apoyo "sostienen" el cojín y evitan que siga expandiéndose. □

Ejercicio 8.3

Para datos separables que contienen ejemplos positivos y negativos, y un hiperplano de separación h, defina el margen del lado positivo *p<(h) como* la distancia entre // y el punto de datos más cercano de la clase -1-1. Del mismo modo, defina el margen del lado negativo p- (//) como la distancia entre h y el punto de datos más cercano de la clase -1. Del mismo modo, defina el margen del lado negativo *p- (//)* como la distancia entre h y el punto de datos más cercano de la clase -1. Argumentar que si // es el hiperplano óptimo, entonces *p+ ft) == pm ft). Es* decir, el grosor del cojín a cada lado del óptimo h es igual.

Hacemos una observación importante que será útil más adelante. En el ejemplo 8.2, ¿qué ocurre con el hiperplano óptimo si eliminamos el punto de datos (iv), el vector sin soporte? Nada. El hiperplano sigue siendo un separador con el *mismo* margen. Aunque hayamos eliminado un punto de datos, no se puede conseguir un margen mayor, ya que todos los vectores de soporte que antes impedían la expansión del margen siguen estando en los datos. Por tanto, el hiperplano sigue siendo óptimo. De hecho, para calcular el hiperplano óptimo, sólo se necesitan los vectores de soporte; los demás datos pueden desecharse.

Programación **cuadrática (**QP). Para conjuntos de datos más grandes, resolver manualmente el problema de optimización en (8.4) como hicimos en el Ejemplo 8.2 ya no es factible.

La buena noticia es que (8.4) pertenece a una familia bien estudiada de problemas de optimización conocidos como programación cuadrática (QP). Siempre que minimice una función *cuadrática* (convexa), sujeta a restricciones de desigualdad de lino, puede utilizar la programación cuadrática. La programación cuadrática es un área tan estudiada que existen excelentes solucionadores disponibles públicamente para muchas plataformas de cálculo numérico. No necesitaremos saber cómo resolver un problema de programación cuadrática; sólo necesitaremos saber cómo tomar cualquier problema dado y convertirlo en una forma estándar que luego se introducirá en un solucionador Q P. Comenzaremos describiendo la forma estándar de un problema de programación cuadrática. Empezaremos describiendo la forma estándar de un problemaQ P:

$$\underset{\mathbf{u}\in\mathbb{R}^{L}}{\text{minimizar:}} \quad \mathbf{u'Qu+p'u} \tag{8.5}$$

sujeto a: apu cp (m = 1, , M).

Las variables a optimizar son los componentes del vector u que tiene dimensión *L*. Todos los demás parámetros Q, p, ay y cp para rn = 1, ... , M son especificados por el usuario. *M* son especificados por el usuario. La cantidad que se minimiza contiene un término cuadrático
2 u*Qu y un término lineal p*u. Los coeficientes de los términos cuadráticos son en la matriz *L* x Q: u*Qu ;_ _ q, mu . Los coeficientes de los términos lineales están en el vector *L* x 1 p: p*u ; pm,. Hay *M* restricciones de desigualdad lineales, cada una especificada por un vector *L* 1 ay y un escalar cp . Para que el problema QP sea convexo, la matriz Q debe ser semidefinida positiva. Es más conveniente especificar los ay como filas de una matriz A de *M* x ñ y los cp como componentes de un vector c de *M* x 1:

$$A = \begin{bmatrix} \text{a-} \\ \mathbf{a}_M^{\mathrm{T}} \end{bmatrix} \qquad C = \begin{bmatrix} \text{c} \\ c_M \end{bmatrix}$$

Utilizando la representación matricial, el problema Q P de (8.5) se escribe como

$$\begin{aligned} &\text{minm ze:} && \$u\ "Qu + p\ "u \\ &\text{sujeto a:} && Un \quad c. \end{aligned} \tag{86}$$

Las *M* restricciones de desigualdad están todas contenidas en la única restricción vectorial Au > c (que debe cumplirse para cada componente). Escribiremos

$$u'+-QP(\ ,p,A,c$$

para denotar el proceso de ejecutar un solucionador Q P en la entrada (Q, p, A, c) para obtener una solución óptima u* para (8.6).[2]

[2] Un comentario rápido sobre la entrada a Q P-solvers. Algunos Q P-solters tomar la desigualdad inversa restricciones Au c , que simplemente significa que usted necesita para negar A y c en (8,6) . Una restricción de igualdad son dos restricciones de desigualdad (o = b n > ò y n ò). Algunos solucionadores aceptan restricciones de cota superior e inferior separadas en cada componente de u. Todas estas variantes diferentes de la programación cuadrática pueden representarse mediante el problema estándar con restricciones de desigualdad lineales en (8.6) .

Sin embargo, los tipos especiales de restricciones como la igualdad y las restricciones de cota superior/inferior a menudo se pueden manejar de una manera numéricamente más estable que la restricción de desigualdad lineal general.

Demostramos ahora que nuestro problema de optimización en (8.4) es efectivamente un problema Q P-. Para ello, debemos identificar Q, p, A y c. En primer lugar, observemos que las cantidades que hay que resolver (las variables de optimización) son (h, w); juntándolas en u, identificamos la variable de optimización u *C$d+'*.

La dimensión del problema de optimización es ñ = *d* -I- 1. La cantidad que estamos minimizando es w*w. Tenemos que escribirlo en la forma u*Qu -I- p*u. Obsérvese que

$$\mathbf{w}^{\mathrm{T}}\mathbf{w} = \begin{bmatrix} b & \mathbf{w}^{\mathrm{T}} \end{bmatrix} \begin{matrix} 0 & Od \\ d & \mathrm{Id} \end{matrix} \begin{matrix} b \\ " \end{matrix} = \mathbf{u}^{\mathrm{T}} \begin{matrix} 0 & 0\mathrm{d\ u} \\ d & \mathrm{Id} \end{matrix},$$

donde *Id* es la matriz *d s d* 'identidad y *Od* el vector cero d-dimensional. Podemos identificar el término cuadrático con Q - $\begin{matrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathrm{d\ I} & \end{matrix}$, y no hay término lineal, por lo que p = *Od+i* . *En* cuanto a las restricciones de (8.4), hay *N* de ellas en (8.4), por lo que *M - N.* La n-ésima restricción es pq (w*xp -I- fi) **1** , que equivale a

$$\begin{bmatrix} y_n & y_n\mathbf{x}_n^{\mathrm{T}} \end{bmatrix} \mathbf{u} \geq 1,$$

y eso corresponde a poner ap - pq 1 xqJ y cp - 1 en (8.5). Así pues, la matriz A contiene filas que están muy relacionadas con una vieja amiga del capítulo 3, la matriz de datos X aumentada con una columna de 1s. De hecho, la fila n-ésima de A no es más que la fila n-ésima de la matriz de datos pero multiplicada por su etiqueta 9q. El vector de restricciones c = Ip, un vector N-dimensional de unos.

Ejercicio 8.4

SeaY una matriz diagonal *N x N* con entradas diagonales Ynn //n (una versión matricial del vector objetivo y). SeaX la matriz de datos aumentada con una columna de 1s. Demuestre que

$$\mathrm{A} = \mathrm{YX}.$$

A continuación resumimos el algoritmo para obtener un hiperplano óptimo, que se reduce a un problema QP en *d* -I- 1 variables con *N* restricciones.

SVM lineal de margen duro con QP

¡1: Sea p = *Od+! (d* -I- 1)-vector cero dimensional) y c ly (vector N-dimensional de unos). Construir las matrices Q y A, donde

$$\mathrm{Q} = \begin{matrix} 0 & Od \\ d & \mathrm{Id} \end{matrix} \qquad \mathrm{A} \begin{matrix} y_1 & —y_1\mathbf{x}_1^{\mathrm{T}}— \\ y_N & —y_N\mathbf{x}_N^{\mathrm{T}}— \end{matrix}$$

matriz de datos con signo

2: Calcular = u* +- QP(Q, p, A, c).

3: Devuelve la hipótesis g(x) = sign(w*"x -I- b*).

El modelo de aprendizaje que hemos estado analizando se conoce como *vector de soporte* lineal de margen duro (SVM lineal de margen duro). Sí, suena como algo sacado de un libro de ciencia ficción . No se preocupe, es sólo un nombre que describe un algoritmo para construir un modelo lineal óptimo, y en este algoritmo sólo algunos de los puntos de datos tienen un papel que desempeñar en la determinación de la hipótesis final - estos pocos puntos de datos se denominan *vectores de apoyo,* como ya se ha mencionado. Que el margen sea duro significa que no se permite que ningún dato quede dentro del cojín. Podemos relajar esta condición, y lo haremos más adelante en este capítulo.

Ejercicio 8.5

Demuestre que la matriz @ descrita en el algoritmo SVM lineal de margen duro anterior es semidefinida positiva (es decir, u"@u > 0 para cualquier u).

El resultado significa que el problema QP es convexo. La convexidad es útil porque hace "fácil" encontrar una solución óptima. De hecho, los solucionadores QP estándar pueden resolver nuestro problema QP convexo en *O((N d)').*

Ejemplo 8.3. Podemos construir explícitamente el problema Q P para nuestro ejemplo de juguete del Ejemplo 8.2. Construimos Q, p, A y c como sigue Construimos Q, p, A y c como sigue.

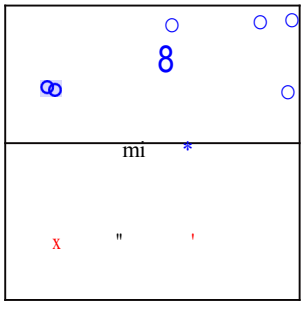
$$Q = \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & \\ 0 & 0 & 1 \end{matrix} \qquad 0p = \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \qquad A \begin{matrix} -1 & 0 & 0 \\ -1 & -2 & -2 \\ 1 & 2 & 0 \\ 1 & 3 & 0 \end{matrix} \qquad c \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

Un solucionador Q P estándar da (h*, in*, m_2) (-1, 1, -1), la misma solución que calculamos manualmente, pero obtenida en menos de un milisegundo. O

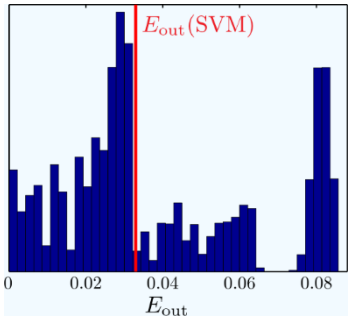
Además de los solucionadores QP estándar que están disponibles públicamente, también hay solucionadores específicamente adaptados para la SVM lineal, que a menudo son más eficientes para el aprendizaje a partir de datos *a gran escala*, caracterizados por un gran *N* o *d.* Algunos paquetes se basan en una versión del descenso de gradiente estocástico (SGD), una técnica de optimización que introdujimos en el Capítulo 3, y otros paquetes utilizan técnicas de optimización más sofisticadas que aprovechan las propiedades especiales de la SVM (como la redundancia de los vectores sin soporte).

Ejemplo 8.4 (Comparación de SVM con PLA). Construimos un conjunto de datos de juguete y lo utilizamos para comparar la SVM con el algoritmo de aprendizaje perceptrón. Para generar los datos, generamos aleatoriamente zt 0,y -1, 1) con la función objetivo siendo -I-1 sobre el eje zt; J(x) = sign(+2) En este experimento, utilizamos la versión de PLA que actualiza los pesos utilizando el punto mal

clasificado xp con menor índice n. El histograma de la Figura 8.1 muestra lo que suele ocurrir con PLA. Dependiendo del orden de los datos, PLA a veces puede tener suerte y vencer a la SVM, y a veces puede ser mucho peor. El clasificador SVM (margen máximo) no depende del orden (aleatorio) de los datos. O



(a) Separador de datos y SVM



(b) Histograma de fi "i (PLA)

Figura 8.1: (a) El clasificador SVM a partir de datos generados utilizando /(x) - sign(+z) (la región azul es /(x) = +1). El margen (cojín) está dentro de las líneas grises y los tres vectores de soporte están encerrados en recuadros. (b) Histograma de *Eg* para los clasificadores PLA que utilizan ordenaciones aleatorias de los datos.

Ejercicio 8.6

Construya un conjunto de datos de juguete con *N* -- 20 utilizando el método del Ejemplo 8.4.

(a) Ejecute el algoritmo SVM para obtener el separador de máximo margen (,)SvM y calcule su Aout y margen.

(b) Construya un ordenamiento de los puntos de datos que resulte en un hiperplano con mal *T: "i* cuando se ejecuta PLA sobre él. [*Pista: Identifique un punto de datos positivo y uno negativo Tor cuya mediatriz* que separa *estos dos puntos sea un mal separador. ¿Dónde deberían estar estos dos puntos en la ordenación? ¿Cuántas iteraciones tardará PLA)?*

(c) Cree un gráfico de sus dos separadores derivados de SVM y PLA.

8.1.2 ¿Es mejor un separador de grasas?

La SVM (hiperplano óptimo) es un modelo lineal, por lo que no puede ajustarse a ciertas funciones simples, como se discutió en el Capítulo 3. Pero, en el lado positivo, la SVM también hereda la buena capacidad de generalización del modelo lineal simple, ya que la dimensión VC está limitada por *d* -I- 1. ¿Gana la máquina de vectores de soporte más capacidad de generalización maximizando el margen, proporcionando un colchón de seguridad por así decirlo? Ahora que tenemos el algoritmo que nos da el hiperplano óptimo, estamos en condiciones de arrojar algo de luz sobre esta cuestión.

Ya hemos argumentado intuitivamente que el hiperplano óptimo es robusto al ruido. Ahora mostramos este vínculo con la regularización de forma más explícita. Hemos visto antes un problema de optimización similar al de (8.4). Recordemos que el soft-

cuando hablamos de la regularización en el capítulo 4,

minimizar: en ( )

sujeto a: w "w C.

En última instancia, esto condujo a la regularización por decaimiento del peso. En la regularización, minimizamos in dado un presupuesto C para w*w. Para el hiperplano óptimo (SVM), minimizamos w*w sujeto a un presupuesto en in (es decir, in ' 0). En cierto sentido, el hiperplano óptimo está haciendo una regularización automática de caída de peso.

	hiperplano óptimo	regularización
minimizar:	w*w	Eq
sujeto a:		w*w ≤ C

Ambos métodos intentan ajustar los datos con pesos pequeños. Esa es la esencia de la regularización. Este vínculo con la regularización es interesante, pero ¿sirve de algo? Sí, tanto en la teoría como en la práctica. Esperamos convencerte de tres cosas.

(a) En la práctica, un separador con márgenes más grandes da mejores resultados. Lo ilustraremos con un sencillo experimento empírico.

(b) Los hiperplanos gordos generalizan mejor que los hiperplanos finos. Demostraremos esto acotando el número de puntos que los hiperplanos gordos pueden destrozar. Nuestro límite es menor que *d* -I- 1 para hiperplanos suficientemente gordos.

(c) El error fuera de la muestra puede ser pequeño, incluso si la dimensión d es grande. Para demostrarlo, acotamos el error de validación cruzada fi" (recordemos que A es un sustituto de fiqqt). Nuestro límite no depende explícitamente de la dimensión.

(i) Un margen mayor es mejor. Utilizaremos el problema de aprendizaje de juguete del Ejemplo 8.4 para estudiar el rendimiento de los separadores con márgenes diferentes. Generamos un conjunto de datos separables de tamaño N = 20. Hay infinitos hiperplanos de separación. Hay infinitos hiperplanos de separación. Muestreamos estos hiperplanos de separación aleatoriamente.

Para cada hiperplano de separación aleatorio fi, podemos calcular A..t(b) y el margen *p(h)/* p(5VM) (normalizado por el margen máximo posible alcanzado por la SVM) . Ahora podemos trazar cómo Aqqt (fi) depende del jrnc/ion del margen disponible que utiliza fi. Para cada conjunto de datos, muestreamos 50.000 hiperplanos de separación aleatorios y luego promediamos los resultados sobre varios miles de conjuntos de datos. La figura 8.2 muestra la dependencia de Appt frente al margen.

La figura 8.2 sugiere claramente que, al elegir un hiperplano de separación aleatorio, es mejor elegir uno con mayor margen. La SVM, que elige tfie

hiperplano con mayor margen, está haciendo un trabajo mucho mejor que uno de los (típicos) hiperplanos de separación al azar. Tenga en cuenta que una vez que llegue a un margen suficientemente grande, el aumento del margen más puede perjudicar fiqqt: es posible mejorar ligeramente el rendimiento entre los hiperplanos al azar sac- rificando ligeramente en el margen máximo y tal vez mejorar de otras maneras. Construyamos ahora algunas pruebas teóricas de por qué es buena la separación con márgenes grandes.

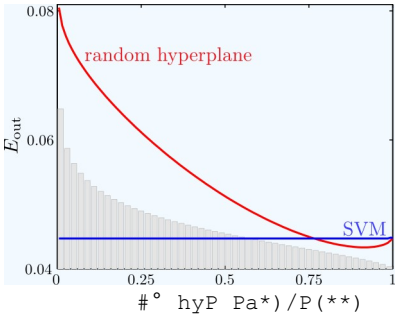


Figura 8.2: Dependencia deñ "i frente al margen *p* para un hiperplano de separación aleatorio. El histograma sombreado del fondo muestra las frecuencias relativas de un margen concreto al seleccionar un hiperplano aleatorio.

(ii) Los hiperplanos gordos destrozan menos puntos. La dimensiónVC de un conjunto de hipó- tesis es el número máximo de puntos que se pueden destrozar. Una dimensión VC más pequeña da una barra de error de generalización más pequeña que une,q .t (ver Capítulo 2 para más detalles). Consideremos el conjunto de hipótesis*p* que contiene *todos los* hiperplanos gordos de anchura (margen) al menos p. Una dicotomía sobre un conjunto de datos (xj , p) ,, (xy , py) puede ser implementada por una hipótesis fi *Op* si % fi(xp) y ninguno de los xp se encuentra dentro del margen de fi. Supongamos que *dzz (p)* es el número máximo de puntos que p Attn rompe. Nuestro objetivo es demostrar que la restricción del conjunto de hipótesis a los separadores de grasa puede disminuir el número de puntos que pueden romperse, es decir, *dpz p)* *d* -I- 1. He aquí un ejemplo.

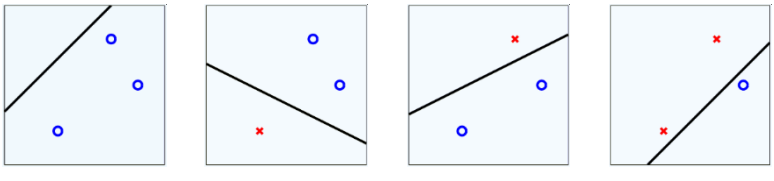


Figura 8.3: Los hiperplanos finos pueden implementar las 8 dicotomías (las otras 4 dicotomías se obtienen negando los pesos y el sesgo).

Los separadores finos (grosor cero) pueden romper los tres puntos mostrados anteriormente. A medida que aumentemos el grosor del separador, pronto no podremos aplicar la dicotomía más a la derecha. Finalmente, a medida que aumentamos el grosor aún

3Técnicamente, una hipótesis en Up no es una clasificación porque dentro de su rrrargen la salida no está definida. No obstante, aún podemos calcular el número máximo de puntos que pueden romperse y esta "dimensión VC" desempeña un papel similar al de dv¿ a la hora de establecer una barra de error de generalización para el modelo, aunque utilizando un análisis más sofisticado.

más, las únicas dicotomías que podremos aplicar serán las dicotomías constantes.

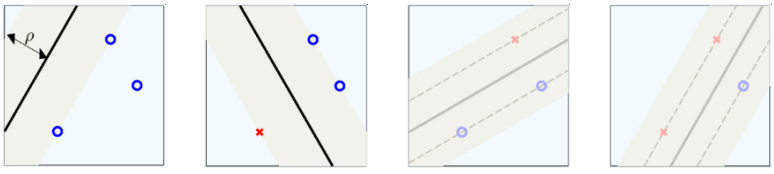


Figura 8.4: Sólo 4 de las 8 dicotomías pueden separarse mediante hiperplanos de grosor *p*. Las líneas discontinuas muestran el separador más grueso posible para cada dicotomía no separable.

En la Figura 8.4, no hay ningún hiperplano p-grueso que pueda separar las dos dicotomías de la derecha. Este ejemplo ilustra por qué los hiperplanos gruesos implementan menos de las dicotomías posibles. Observe que los hiperplanos sólo "parecen" gruesos porque los datos están muy juntos. Si separáramos más los datos, incluso un hiperplano grueso podría aplicar todas las dicotomías. Lo que importa es el grosor del hiperplano en relación con la separación de los datos.

Ejercicio 8.7

Supongamos que los datos se limitan a una esfera unitaria.

(a) Demuestre que *d (p)* es no creciente en *p*.

(b) En 2 dimensiones, demuestre que *da p)* 3 ^{3} *[Pista: Muestre elt* para *p* > $_{2}$
Para cualesquiera' 3 puntos en el disco unitario, debe ***haber dos que*** *estén a menos de* 3 *de distancia el uno del otro. Utilice esta Acct para construir o dicotomía la t no puede ser implementado por cualquier p-grueso separo tor].*

El ejercicio muestra que para un espacio de entrada acotado, los separadores gruesos pueden romper menos de *d* -I- 1 puntos. En general, podemos demostrar el siguiente resultado.

Teorema 8.5 (Dimensión VC de los hiperplanos gordos) . Supongamos que el espacio de entrada es la bola de radio fi en i^{d} , por lo que ||x || fi. Entonces,

$$d_{\mathrm{VC}}(\rho) \leq \lceil R^2/\rho^2 \rceil + 1,$$

donde fi /p°² es el menor número entero mayor o igual que fi°/p² .

También sabemos que la dimensión de la CV es como máximo *d* -I- 1,

por lo que podemos elegir el mejor de los dos límites, y ganamos cuando fi/p es pequeño. El hecho más importante de este límite basado en el margen es que no depende explícitamente de la dimensión *d*. Esto significa que si transformamos los datos a un espacio de dimensión alta, incluso infinita, siempre que utilicemos separadores suficientemente gordos, obtendremos

buena generalización. Como este resultado establece un vínculo crucial entre el margen y la buena generalización, damos una demostración sencilla, aunque algo técnica.

> Comienza el salto seguro: La prueba es bonita pero no esencial. Un recuadro verde similar te indicará cuándo reincorporarte.

Proo/. Fijemos xt , ... , xp que están rotos por hiperplanos con margen p. Demostraremos que cuando *N* es par, *N* fi /p22 -I- 1. Cuando *N* es impar, se obtiene un resultado similar pero
Un análisis más cuidadoso (véase el problema 8.8) muestra que N fi /p 122 $\frac{1}{N}$ En ambos casos, N fi /p22 -t- 1.

Supongamos que N es par. Necesitamos el siguiente hecho geométrico (que se demuestra en el problema 8.7). Existe una dicotomía (equilibrada) pt , ... , 9q tal que

$$\sum_{n=1}^{N} y_n = 0, \text{ and } \left\| \sum_{n=1}^{N} y_n \mathbf{x}_n \right\| \leq \frac{NR}{N\text{-}'} \tag{8.7}$$

(Para pq aleatorio, ppxp es un paseo aleatorio cuya distancia al origen no crece más rápido que fi .) La dicotomía que satisface (8.7) se separa con margen al menos *p* (ya que xj , ... , xy se rompe). Entonces, para algún (w, h),

p| w | < pp (w*xp -I- fi), para n - 1, ... , N.

Sumando sobre n, utilizando 9q = 0 y la desigualdad z de Cauchy-Schwar,

$$N\rho\|\mathbf{w}\| \leq \mathbf{w}^{\mathrm{T}} \sum_{n=1}^{N} y_n \mathbf{x}_n + b \sum_{n=1}^{N} y_n = \mathbf{w}^{\mathrm{T}} \sum_{n=1}^{N} y_n \mathbf{x}_n \leq \|\mathbf{w}\| \left\| \sum_{n=1}^{N} y_n \mathbf{x}_n \right\|.$$

Por el límite en (8.7), el RHS es a lo sumo | *w* | *N R/ N* - , o:

$$\rho \leq \frac{R}{\sqrt{N-1}} \implies N \leq \frac{R^2}{\rho^2} + 1.$$

> **Fin del salto seguro:** Bienvenido de nuevo para un resumen.

Combinando el Teorema 8.5 con dpp (p) dgp (0) - d -I- 1, tenemos que

dzz (p) min (fi°/p° , *d)* 1.

El límite sugiere que p puede utilizarse para controlar la complejidad del modelo. El hiperplano de separación (en ' 0 Con el máximo margen p tendrá el menor *dpz(p)* y, por tanto, la menor barra de error de generalización.

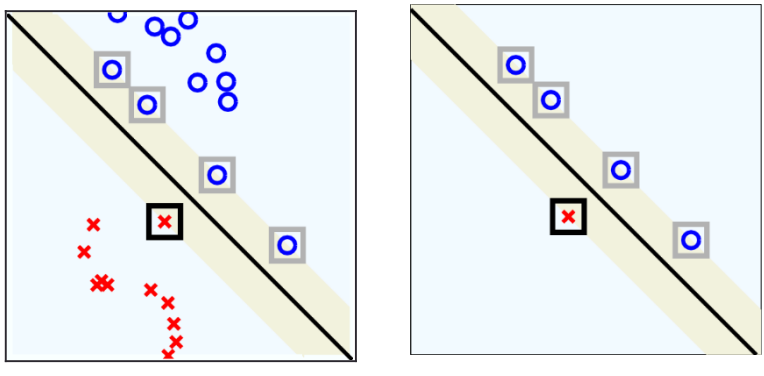
Por desgracia, existe una complicación. No sabemos de antemano cuál es la anchura p correcta (¿y si elegimos una *p* más alta de lo posible?). El algoritmo

del hiperplano óptimo fija p sólo después de ver los datos. Darse la opción de usar un p más pequeño pero decidirse por un p más alto al ver los datos significa que los datos están siendo fisgoneados. Hay que modificar el análisis de la CV para tener en cuenta este tipo de fisgoneo de datos. El lector interesado puede encontrar los detalles relacionados con estos tecnicismos en la bibliografía.

(iii) Limitación **del error de validación cruzada.** En el Capítulo 4, introdujimos el error de validación cruzada de exclusión A, que es la media de los errores de exclusión eg. Cada eg se calcula excluyendo el punto de datos (xp , pp) y aprendiendo sobre los datos restantes para obtener pt . La hipótesis pt se evalúa en (xp, pq) para obtener eg, y

$$E_{\text{cv}} = \frac{1}{N} \sum_{n=1}^{N} \mathsf{e}_n.$$

Una propiedad importante de€ es que es una estimación insesgada del error esperado fuera de la muestra para un conjunto de datos de tamaño N - 1. (véase el capítulo 4 para más detalles sobre la validación). Por lo tanto, A es una aproximación muy útil para fiqqt. Podemos obtener un límite sorprendentemente sencillo para A utilizando el número de vectores de soporte. Recordemos que un vector de soporte se encuentra en el borde del margen del hiperplano óptimo. Ilustramos un conjunto de datos de juguete en la Figura 8.5(a) con los vectores de soporte resaltados en recuadros.



(a) Todos los datos (b) Sólo vectores soporte

Figura 8.5: (a) Hiperplano óptimo con vectores soporte encerrados en recuadros. (b) El mismo hiperplano se obtiene utilizando sólo los vectores de soporte.

La observación crucial, como se ilustra en la Figura 8.5(b), es que si se eliminan cualquiera (o todos) los puntos de datos distintos de los vectores de soporte, el separador resultante producido por la SVM no cambia. En el problema 8.9 se le pide que demuestre esto formalmente, pero la figura 8.5 debería servir como justificación suficiente. Esta observación tiene una implicación importante: eg = 0 para cualquier punto de datos (xq, pp) que no sea un vector soporte. Esto se debe a que la eliminación de ese punto de datos da como resultado el mismo separador, y puesto que (xp , pp) se clasificó correctamente antes de la

eliminación, seguirá siendo correcto después de la eliminación. Para los v e c t o r e s d e soporte, ep 1 (binario

error de clasificación), y así

$$E''(\text{SVM}) \quad \text{soporte} \quad \text{ke}, \leq \frac{\text{vectores de}}{N} \tag{8.8}$$

Ejercicio 8.8

(a) Evalúe el límite en (8.8) **para los datos** de la Figura 8.5.

(b) Si se elimina uno de los cuatro vectores de apoyo de un cuadro gris, ¿cambia el clasificador?

(c) Utiliza tu respuesta en (b) para mejorar tu límite en (a).

Los vectores de soporte de los recuadros grises no son esenciales y el vector de soporte del recuadro negro es esencial. Se puede mejorar el límite de (8.8) para utilizar sólo los vectores de soporte esenciales. El número de vectores de soporte es ilimitado, pero el número de vectores de soporte esenciales es como máximo *d+* 1 (normalmente mucho menor).

En aras de una información completa, y para ser justos con la APA, debemos señalar que también se puede obtener un límite de A" para la APA, a saber

$$\text{A (PLA)} \quad \frac{\text{fi}}{N^{p2}},$$

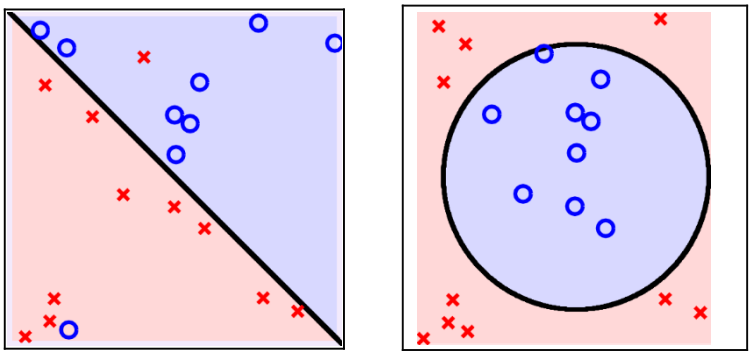
donde *p* es el margen del hiperplano más grueso que separa los datos, y fi es un límite superior de | xp || (véase el problema 8.11). La siguiente tabla proporciona un resumen de lo que sabemos basándonos en lo que hemos discutido hasta ahora.

Algoritmo de selección del hiperplano de separación		
General	PLA	SVM (Hiperplano óptimo)
$d_{\text{VC}} = d + 1$		d (p) min $\left\lceil \frac{R^2}{\rho^2} \right\rceil, d\Big) + 1$
	$E_{\text{cv}} \leq \frac{R^2}{N\rho^2}$	''' $\frac{\text{vectores de apoyo}}{N}$

En general, todo lo que se puede concluir es el límite VC basado en una dimensión VC de *d* -I- 1. En dimensiones altas, este límite puede ser muy flojo. Para PLA o SVM, tenemos límites adicionales que no dependen explícitamente de la dimensión *d.* Si el margen es grande, o si el número de vectores de soporte es pequeño (incluso en dimensiones infinitas), todavía estamos en buena forma.

8.1.3 Datos no separables

Hasta ahora hemos supuesto que los datos son linealmente separables y nos hemos centrado en separar los datos con el máximo colchón de seguridad. ¿Y si los



(a) Datos ruidosos.

(b) No linealmente separable.

Figura 8.6: Datos no separables (reproducción de la figura 3.1).

¿los datos no son linealmente separables? La figura 8.6 (reproducida del capítulo 3) ilustra los dos tipos de no separabilidad. En la Figura 8.6(a), dos puntos de datos ruidosos hacen que los datos no sean separables. En la Figura 8.6(b), la función objetivo es inherentemente no lineal.

Para el problema de aprendizaje de la Figura 8.6(a), preferimos el separador lineal, y necesitamos tolerar los pocos puntos de datos ruidosos. En el Capítulo 3, modificamos el PLA en el algoritmo de bolsillo para manejar esta situación. Del mismo modo, para las SVM, modificaremos la SVM de margen duro a la SVM de margen so -rnorpin en la Sección 8.4. A diferencia de la SVM de margen duro, la SVM de margen blando permite que los puntos de datos violen el cojín, o incluso que se clasifiquen erróneamente.

Para abordar la otra situación de la figura 8.6(b), introdujimos la transformada no lineal en el capítulo 3. Nada nos impide utilizar la transformada no lineal con el hiperplano óptimo, lo que haremos aquí.

Para que los datos sean separables, lo normal es transformar a una dimensión superior. Consideremos una transformación 4' :id :id. Los datos transformados son

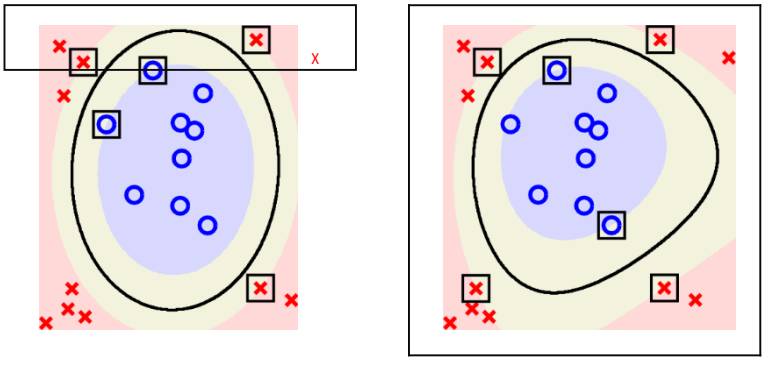
Z - @ (X) .

Después de transformar los datos, resolvemos el problema SVM de margen duro en el espacio I, que es simplemente (8.4) escrito con zp en lugar de xp :

$$\begin{aligned} \text{minimizar:} \quad & \frac{1}{2}\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{w}} \\ \tilde{b},\tilde{\mathbf{w}} \\ \text{subject to:} \quad & y_n\left(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{z}_n + \tilde{b}\right) \geq 1 \qquad (n = 1, \cdots, N), \end{aligned} \tag{8.9}$$

donde w está ahora en R en lugar *de* $\tilde{\boldsymbol{i}}^{d}$ (recordemos que usamos tilde para objetos en el espacio I

). El problema de optimización de (8.9) es un problema Q P con *d* 1



(a) SVM Con z (b) SVM con 43

Figura 8.7: Separación no lineal utilizando la SVM con transformadas polinómicas de 2º y 3er orden. El margen está sombreado en amarillo y los vectores de soporte están en recuadro. La dimensión de &3 es casi el doble que la de 'b2, pero el separador SVM resultante no se sobreajusta gravemente con 4'3.

variables de optimización y N restricciones. Para resolver este problema de optimización, podemos utilizar el algoritmo estándar de margen duro del cuadro de algoritmos de la página 8- 10, después de sustituir xp por zq y *d* por *d.* El algoritmo devuelve una solución óptima b*, w* y la hipótesis final es

$$g(\mathbf{x}) = \text{sign}(\tilde{\mathbf{w}}^{*\text{T}}\Phi(\mathbf{x}) + b^*).$$

En el capítulo 3, introdujimos una transformada polinómica general de k-ésimo orden 4',. Por ejemplo, la transformada polinómica de segundo orden es

$$\Phi_2(\mathbf{x}) = (x_1, x_2, x_1^2, x_1x_2, x_2^2).$$

La figura 8.7 muestra el resultado de utilizar la SVM con las transformadas polinómicas de 2º y 3er orden 2 3 para los datos de la Figura 8.6(b). Obsérvese que el "margen" no tiene una anchura constante en el espacioI es en el espacioI donde la anchura del separador es uniforme. Además, en la figura 8.7(b) se puede ver que algunos puntos azules cerca de la parte superior izquierda no son vectores de apoyo, a pesar de que parecen estar más cerca de la superficie de separación que los vectores de apoyo rojos cerca de la parte inferior derecha. De nuevo, esto se debe a que lo que importa son las distancias al separador en el espacio I. Por último, las dimensiones de los dos espacios de características son d2 = y *d z* - 9 (casi el doble para la transformada polinómica de 3er orden). Sin embargo, el número de vectores de soporte aumentó de 5 a sólo 6, por lo que el límite de A no se duplicó prácticamente.

La ventaja de la transformada no lineal es que podemos utilizar una sofisticada

límite inferior i-n Sin embargo, hay que pagar un precio para conseguir esta sofisticación.

en términos de una mayor dp y una tendencia a sobreajustar. Esta compensación se destaca en la tabla siguiente para PLA, en comparación con SVM.

	perceptrones	perceptrón + transformación
complexity control	lineal pequeño d_{VC} dpq	large d_{VC} grande
límite	lineal	sofisticado

Observamos en la Figura 8.7(b) que la SVM polinómica de 3er orden no muestra el nivel de sobreajuste que cabría esperar cuando casi duplicamos el número de parámetros libres. Podemos tener nuestro pastel y comérnoslo también: disfrutamos del beneficio de las transformaciones de alta dimensión en términos de obtener límites sofisticados y, sin embargo, no pagamos demasiado en términos de A..t porque *dpz* y A se pueden controlar en términos de cantidades no directamente vinculadas a *d.* La tabla que ilustra las compensaciones para la SVM es:

	SVM	SVM + transformación
complexity control	lineal smaller '*dpt*' effective d_{VC}	*dpt* "effective" large d_{VC} grande
límite	lineal	sofisticado

Ahora tienes la máquina de vectores soporte a tu alcance: un modelo lineal muy potente y fácil de usar que viene con regularización automática. Podríamos haber terminado, pero en lugar de eso, vamos a presentarle una herramienta muy poderosa que se puede utilizar para implementar transformaciones no lineales llamada kernel. El kernel le permitirá explotar al máximo las capacidades de la SVM. La SVM tiene una robustez potencial al sobreajuste incluso después de transformarse a una dimensión mucho mayor que abre un nuevo mundo de posibilidades: ¿qué pasa con el uso de transformaciones de dimensión infinita? Sí, ¡infinitas! Ciertamente, con nuestra tecnología actual no es factible utilizar una transformación de dimensión infinita; pero, utilizando el "truco del núcleo", no sólo podemos hacer factibles las transformaciones de dimensión infinita, sino que también podemos hacerlas eficientes. Estén atentos, es un tema apasionante.

8.2 Doble formulación de la SVM

La promesa de infinitas transformaciones no lineales dimensionales sin duda abre el apetito, pero vamos a tener que ganar nuestra galleta . Vamos a introducir el problema dual de SVM que es equivalente al problema *primal* original (8.9) en que la solución de ambos problemas da el mismo hiperplano óptimo, pero el problema dual es a menudo más fácil. Es esta visión dual (sin juego de palabras) de la SVM la que nos permitirá explotar el truco del kernel que tanto hemos pregonado. Pero la visión dual también es una fórmula importante de la SVM que nos dará más información sobre el hiperplano óptimo. La conexión entre los problemas de optimización primal y dual es un tema muy estudiado en la teoría de la optimización, y sólo introducimos las pocas piezas que necesitamos para la SVM.

Obsérvese que (8.9) es un problema Q P con *d* -I- 1 variables (h, w) y N contensiones. Resulta difícil desde el punto de vista computacional resolver (8.9) cuando *d* es grande, por no hablar de

infinito. El problema dual también será un problema Q P, pero con *N* variables y *N* - I- 1 restricciones la carga computacional ya no depende de d, lo que supone un gran ahorro cuando pasamos a *d* muy altos.

Derivar el dual no va a ser fácil, pero en el lado positivo, ya hemos visto la herramienta principal que vamos a necesitar (en la Sección 4.2 cuando hablamos de la regularización). La regularización nos introdujo en el problema de minimización restringida

$$\text{minimizar: Aip (w) sujeto}$$

$$\text{a: } \mathbf{w} \quad '' \text{ w U ,}$$

donde C es un parámetro especificado por el usuario. Demostramos que para un C dado, existe un A tal que la minimización del error aumentado fi,qg (w) = Aip (w) + Aw*w da la misma solución regularizada. Consideramos el *multiplicador* £npronpe *5* como el parámetro especificado por el usuario en lugar de U, y minimizamos el error aumentado en lugar de resolver el problema de minimización restringido. Aquí, también vamos a utilizar los multiplicadores de Lagrange, pero de una manera ligeramente diferente. Los multiplicadores de Lagrange surgirán como roria#/es que corresponden a las restricciones, y necesitamos formular un nuevo problema de optimización (que es el problema dual) para resolver para esas variables.

8.2.1 Dual de Lagrange para un problema QP

Ilustremos el concepto de dual con una versión simplificada del problema Q P estándar, utilizando sólo una restricción de desigualdad. Todos los conceptos se generalizarán fácilmente al caso con más de una restricción. Consideremos el problema Q P

$$\text{minimizar:} \qquad \text{u "Qu -I- p} \tag{8.10}$$

$$\text{"u sujeto a:} \qquad \text{a "u} \quad \text{c}$$

He aquí un problema de optimización estrechamente relacionado.

$$\underset{\mathbf{u}\in\mathbb{R}^{L}}{\text{minimizar:}} \qquad \text{y u*Qu -I- p*u -I- max n (c - a*u) .} \tag{8.11}$$

La variable n 0 multiplica la restricción (c- a*u).[4] Para obtener el objetivo en (8.11) a partir de (8.10), añadimos lo que equivale a un término de penalización que fomenta que (c - a*u) sea negativo y satisfaga la restricción. El problema de optimización de (8.10) es equivalente al de (8.11) siempre que haya al menos una solución que satisfaga la restricción de (8.10). La ventaja en (8.11) es que la minimización con respecto a u *no tiene restricciones,* y el precio es un objetivo ligeramente más complejo que implica este "término de penalización lagrangiano". El siguiente ejercicio demuestra la equivalencia.

[4] El parámetro cr se denomina multiplicador de Lagrange. En la literatura de optimización, el multiplicador de Lagrange se denota típicamente por A. Históricamente, la literatura SVM ha utilizado ct, que es la convención que seguimos.

Ejercicio 8.9

Sea uo óptimo para (8.10), y sea ui óptimo para (8.11).

(a) Demuestre que puede a (c - a "uo) 0. *Pista: c* - a "uo 0./

(b) Demuestre que ui es factible para (8.10). Para demostrarlo, supongamos por el contrario que c- a "u1 > 0. Demostremos que el objetivo en (8.11) es infinito, mientras que uo alcanza un objetivo finito de 2u#@uo -I- p "uo, lo que contradice la optimalidad de ui .

(c) Demuestre que $\frac{1}{2}\mathbf{u}_1^T Q\mathbf{u}_1 + \mathbf{p}^T\mathbf{u}_1 = \frac{1}{2}$ u#@uo -I- p "uo, y por tanto que ui es óptimo para (8.10) y uo es óptimo para (8.11).

(d) Sea u" cualquier solución óptima de (8.11) con mayo a(c - a "u") alcanzada en a* Demuéstrese que

$$a^*(-c \quad a"u^*) \quad 0. \qquad (8.12\}$$

O bien la restricción se cumple exactamente con -c a "u "0, o bien a" 0.

El ejercicio 8.9 muestra que siempre que el problema original en (8.10) sea factible, podemos obtener una solución resolviendo (8.11) en su lugar. Analicemos (8.11) con más detalle. Nuestro objetivo es simplificarlo. Introduzcamos la función lagrangiana 6(u, n), definida por

$$£(u, n) = {}_2 u "Qu + p "u -I- n (c - a^*u) \qquad (8.13)$$

En términos de 6, el problema de optimización en (8.11) es

$$\min \max £(u, n). \qquad (8.14)$$

Para la programación cuadrática convexa, cuando 6(u, o) tiene la forma especial de (8.13) y c - a*u 0 es factible, se ha demostrado que se cumple una relación profunda conocida como s/ronp *dualil y*:

$$\text{minmBxL(u,D)} = \text{mBxminL(u,D)}. \qquad (8.15)$$

El lector interesado puede consultar una referencia estándar en optimización convexa para obtener una prueba. El impacto para nosotros es que una solución al problema de optimización en el lado derecho de (8.15) da una solución al problema en el lado izquierdo, que es el problema que queremos resolver. Esto nos ayuda porque en el lado derecho, primero se está minimizando con respecto a una u *sin restricciones,* y eso lo podemos hacer analíticamente. Este paso analítico reduce considerablemente la complejidad del problema. Resolver el problema en el lado derecho de (8.15) se conoce como resolver el *problema dual de Lagrange* (problema dual para abreviar). El problema original se denomina problema primal.

Discutimos brevemente qué hacer cuando hay muchas restricciones, apu cp para m = 1, ... , *M.* No hay muchos cambios. Todo lo que hacemos es introducir

un Lagrange

multiplicador eg 0 para la restricción eocfi y añadimos el término de penalización up(cp - apu) en el Lagrangiano. A continuación, igual que antes, resolvemos el problema dual. Un ejemplo sencillo ilustrará toda la mecánica.

Ejemplo 8.6. Minimicemos $u_1^2+u_2^2$ sujeto a u -I- 2-z > 2 y u ,> 0. Primero construimos el Lagrangiano,

$$\mathcal{L}(u,o) - u_1^2+u_2^2+\alpha_1(2-u_1-2u_2)-\alpha_2u_1-\alpha_3u_2,$$

donde, como se puede ver, hemos añadido un término de penalización para cada una de las tres restricciones, y cada término de penalización tiene un multiplicador de Lagrange asociado. Para resolver el problema de optimización dual, primero tenemos que minimizar 6(u, n) con respecto a la u sin restricciones. A continuación, tenemos que maximizar con respecto a n 0. Para hacer la minimización sin restricciones con respecto a u, utilizamos la condición estándar de la primera d$\frac{\partial\dot{\mathcal{L}}}{\partial u_1}$ ' del cálculo:

$$\frac{\partial\mathcal{L}}{\partial u_1}=0 \quad\Longrightarrow\quad u_1=\frac{\alpha_1+\alpha_2}{\quad};$$

$$\frac{\partial\mathcal{L}}{\partial u_2}=0 \qquad .2 \quad \frac{20 \quad 3}{2} \tag{-}$$

Si volvemos a introducir estos valores para u y2 en £ y juntamos los términos, tenemos una función sólo de cr, que tenemos que maximizar con respecto a n 2 3:

$$\text{maximizar:} \quad \text{€(a)} - -\frac{52}{\text{¡ 4 ' !}} \quad \frac{1\ 2}{4\ '\ 2} \quad \frac{1\ 2}{4\ '3} \quad \frac{1}{2}\alpha_1\alpha_2-\alpha_1\alpha_3+2\alpha_1$$

$$\text{subject to:} \quad \alpha_1,\alpha_2,\alpha_3\geq 0$$

Ejercicio 8.10

Haz el álgebra. Deduce (+) e introdúcelo en C(u, a) para obtener C(a).

Al pasar al dual, ¡lo único que hemos conseguido es obtener otro problema QP! Así que, en cierto sentido, no hemos resuelto el problema en absoluto. ¿Qué hemos ganado? El nuevo problema es más fácil de resolver. Esto se debe a que las restricciones del problema dual son simples (n O). En nuestro caso, todos los términos que implican 2 y 3 son como máximo cero, y podemos llegar a cero fijando n2 = 3 = 0. Esto nos deja con -4 o2 -I- 2n , que se maximiza cuando n $\frac{4}{5}$. Por tanto, la solución final es α_1 $\frac{4}{5}$, >z - 3 - 0, e introduciendo esto en (+) se obtiene

$$u_1=\frac{2}{5} \qquad \text{y} \qquad u_2=\frac{4}{5}.$$

El valor óptimo del objetivo e s ² $u_1^2+\circ z^2-\frac{4}{5'}$ □

Resumimos nuestra discusión en el siguiente teorema, que establece la formulación dual de un problema Q P . En la siguiente sección, vamos a mostrar

cómo esta formulación dual se aplica a la SVM Q P-problema. El teorema parece formidable, pero no se deje intimidar. Su aplicación a la SVM Q P-problema no es conceptualmente diferente de nuestro pequeño ejemplo anterior.

Teorema 8.7 (KKT). Para un problema Q P convexo factible en forma *primitiva,*

$$\begin{aligned} &\text{minimizar:} && \tfrac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} \\ &\text{sujeto a:} && \mathbf{a}_m^T\mathbf{u} \geq c_m \qquad (m = 1, \cdots, M), \end{aligned}$$

definir la función de Lagrange

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{u}^T Q\mathbf{u} + \mathbf{p}^T\mathbf{u} + \sum_{m=1}^{M} \alpha_m (c_m - \mathbf{a}_m^T\mathbf{u})$$

La solución u* es óptima para el problema primal si y sólo si (u*, ri*) es una solución del problema de optimización dual

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha})$$

El óptimo (u*, ri*) satisface las condiciones de Karush-Kühn-Tucker (KKT):

(i/ *Restricciones primarias y de reparto:*

$$\mathbf{a}_m^T\mathbf{u}^* \geq c_m \qquad \text{para } m = 1, \cdots, M.$$

(iis *Complementar¡j holgura:*

$$\alpha_m^* (\mathbf{a}_m^T\mathbf{u}^* - c_m) = 0$$

(iii) *Stationarity with respect to* **u**:

$$\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}^*, \boldsymbol{\alpha}^*) = 0$$

Las tres condiciones KKT del teorema 8.7 caracterizan la relación entre los óptimos u* y ri* y a menudo pueden utilizarse para simplificar el problema dual de Lagrange. Las restricciones se heredan de las restricciones en las definiciones de los problemas de optimización primal y dual. La holgura complementaria es una condición que derivó en (8.12) que dice que en la solución óptima, las restricciones se dividen en dos tipos: las que están en el límite y se satisfacen exactamente (las restricciones octies) y las que están en el interior del conjunto factible. Las restricciones interiores deben tener multiplicadores de Lagrange iguales a cero. La estacionariedad con respecto a u es la condición necesaria y suficiente para que un programa convexo resuelva la primera parte del problema dual, es decir, ming 6(u, ri). A continuación, nos centramos en la SVM de margen duro y utilizamos las condiciones KKT, en particular, la estacionariedad con respecto a u, para simplificar el problema dual de Lagrange.

8.2.2 Doble de la SVM de margen duro

Ahora aplicamos el teorema KKT al problema convexo Q P para SVM de margen duro (8.9). La mecánica de nuestra derivación no es más compleja que

en

Ejemplo 8.6, aunque el álgebra es más engorrosa. Los pasos que seguimos en el Ejemplo 8.6 son una guía útil a tener en cuenta.

El problema de optimización de margen duro sólo se aplica cuando los datos son separables linealmente. Esto significa que el problema de optimización es convexo y factible, por lo que se aplica el teorema KKT. Para su comodidad, reproducimos aquí el problema QP de SVM de margen duro,

$$\begin{aligned} &\underset{b,\mathbf{w}}{\text{minimizar:}} && \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} && (8.16)\\ &\text{sujeto a:} && \text{pp (w*xp -I- fi)} \quad 1 \qquad (\text{n - 1}, \quad ,\text{N}). \end{aligned}$$

La variable de optimización es u =) . La primera tarea consiste en construir el La- grangiano. Hay N restricciones, por lo que añadimos N términos de penalización e introducimos un multiplicador de Lagrange oq para cada término de penalización. El lagrangiano es

$$\mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) \qquad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)\right)$$

WWW

w'w- G qw'xp-# 0q qT G" (8.17

Primero debemos minimizar 6 con respecto a (h, w) y luego maximizar con respecto a n **O** . Para minimizar con respecto a (h, w), necesitamos las derivadas de £. La derivada con respecto a b es simplemente el coeficiente de b porque b aparece linealmente en el Lagrangiano. Para diferenciar los términos en los que interviene w, necesitamos algunas identidades de cálculo vectorial del apéndice de álgebra lineal: pp w "w = 2w y pp w "x = x. El lector puede comprobar ahora que

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum^{N} \alpha_n y_n \qquad \text{and} \qquad \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum^{N} \alpha_n y_n \mathbf{x}_n.$$

Poniendo a cero estas derivadas se obtiene

$$\sum \alpha_n y_n = 0; \tag{8.18}$$

$$\bullet \; ' \; \mathbb{Z}^{\circ} . \; ' \; . \; ^{\mathrm{X}} . \tag{8.19}$$

Ha ocurrido algo extraño (resaltado en rojo), que no sucedió en el Ejemplo 8.6. Después de fijar las derivadas a cero en el Ejemplo 8.6, pudimos resolver para u en términos de los multiplicadores de Lagrange n. Aquí, la condición de estacionariedad para b no nos permite resolver para b directamente . En su lugar, tenemos una restricción que n debe satisfacer en la solución final. No dejes que esto te inquiete. El teorema de KKT nos permitirá recuperar finalmente h . Esta restricción sobre n no es sorprendente. Cualquier elección de n que no satisfaga

(8.18) sería

permitir £ -' -en eligiendo adecuadamente b. Como rnozirnizamos sobre cr, debemos elegir n para satisfacer la restricción y evitar 6 -' -en.

Procedemos como en el Ejemplo 8.6, volviendo a introducir las restricciones de estacionariedad en el Lagrangiano para obtener una función únicamente en términos de n. Observemos que la restricción de (8.18) significa que el término que implica b en el Lagrangiano (8.17) se reduce a cero. Obsérvese que la restricción en (8.18) significa que el término que implica b en el Lagrangiano (8.17) se reduce a cero. Los términos de la lagrangiana (8.17) que implican los pesos w se simplifican cuando utilizamos la expresión de (8.19):

$$\text{i-'-- Z°- --'X-}$$

$$\frac{1}{2}\sum_{\cdot}^{N}\alpha_n y_n \mathbf{x}_n^{\mathrm{T}}\sum_{\cdot}^{N}\alpha_m y_m \mathbf{x}_m - \sum_{\cdot}^{N}\alpha_n y_n \sum_{\cdot}^{N}\alpha_m y_m \mathbf{x}_m^{\mathrm{T}}\mathbf{x}_n$$

$$\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m - \sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m$$

$$\frac{1}{2}\sum_{}^{N}\sum_{}^{N} \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m. \qquad (8.20)$$

Después de utilizar (8.18) y (8.20) en (8.17), el Lagrangiano se reduce a una función más simple de sólo la variable n:

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{}^{N}\sum_{}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m + \sum_{}^{N}\alpha_n.$$

Ahora debemos maximizar 6(n) sujeto a n 0, y no podemos olvidar la restricción (8.18) que heredamos de la condición de estacionariedad sobre b. Podemos equivalentemente minimizar el negativo de £(n), y así tenemos el siguiente problema de optimización a resolver.

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{minimizar:}} \quad \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m - \sum_{n=1}^{N}\alpha_n \qquad (8.21)$$

$$\text{subject to:} \quad \sum_{n=1}^{N} y_n \alpha_n = 0$$

$$\text{o,>0} \qquad \text{(n=l ,N)}$$

Si has llegado hasta aquí, te mereces una palmadita en la espalda por haber sobrevivido a todo el álgebra . El resultado es similar a lo que ocurrió en el Ejemplo 8.6. No hemos resuelto el problema; lo hemos reducido a otro

problema Q P. El siguiente ejercicio le guía a través de los pasos para poner (8.21) en la forma QP estándar.

Ejercicio 8.11

(a) Demuestre que el problema en (8.21) es un problema QP estándar:

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{minimizar:}} \quad \tfrac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \qquad (8.22)$$

$$\text{sujeto a:} \quad \mathbf{A}\boldsymbol{\alpha} \geq \mathbf{0}$$

donde @o y Ao (D para dual) vienen dados por:

$$Q = \begin{bmatrix} y_1 y_1 \mathbf{x}_1^T \mathbf{x}_1 & \cdots & y_1 y_N \mathbf{x}_1^T \mathbf{x}_N \\ y_2 y_1 \mathbf{x}_2^T \mathbf{x}_1 & \cdots & y_2 y_N \mathbf{x}_2^T \mathbf{x}_N \\ \vdots & \ddots & \vdots \\ y_N y_1 \mathbf{x}_N^T \mathbf{x}_1 & \cdots & y_N y_N \mathbf{x}_N^T \mathbf{x}_N \end{bmatrix} \quad \text{y} \quad \mathbf{A} = \begin{bmatrix} \mathbf{y}^T \\ -\mathbf{y}^* \\ \mathbf{I} \end{bmatrix}$$

[Pista: Recuerda que una igualdad corresponde a dos desigualdades).

(b) La matriz @o de coeficientes cuadráticos es] //// x xn.

Demostrar que XIX*, donde X, es la 'matriz de datos con signo',

$$\mathbf{X}_s = \begin{bmatrix} \#!*1 \\ \$2x2 \\ \\ \text{—} y_N \mathbf{x}_N^T \text{—} \end{bmatrix}$$

Por tanto, demuéstrese que @o es semidefinido positivo. Esto implica que el problema QP es convexo.

Ejemplo 8.8. Ilustremos toda la maquinaria de la formulación dual utilizando los datos del Ejemplo 8.2 (el problema de juguete de la Sección 8.1). Para su conveniencia, reproducimos los datos y calculamos Qp, Ap utilizando el ejercicio 8.11:

$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & \\ 3 & 0 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ \\ +1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 8 & -4 & -6 \\ 0 & -4 & 4 & 6 \\ 0 & -6 & 6 & 9 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ahora podemos escribir el problema de optimización en (8.21) y resolverlo manualmente para obtener el óptimo n. Se invita al lector masoquista a hacerlo en el Problema 8.3. En su lugar, podemos utilizar un solucionador Q P para resolver (8.22) en menos de un milisegundo:

a* QP(@m, -1, Am, 0) da a"

1
0

Ahora que tenemos n*, podemos calcular los pesos óptimos utilizando (8.19),

$$\mathbf{w}^* = \sum^{N} y_n \alpha_n^* \mathbf{x}_n = -\tfrac{1}{2} \begin{vmatrix} 0 \\ 0 \end{vmatrix} - \tfrac{1}{2} \begin{vmatrix} 2 \\ 2 \end{vmatrix} + \begin{matrix} 2 & \text{I} \\ 0 & \text{-1} \end{matrix}$$

Como era de esperar, estos son los mismos pesos óptimos que obtuvimos en el Ejemplo 8.2. ¿Qué pasa con b? Aquí es donde las cosas se ponen difíciles. Recordemos la condición KKT de holgura complementaria del teorema 8.7. Afirma que si o 0, entonces la restricción correlativa debe ser 0. Afirma que si o **O** , entonces la restricción correlativa debe satisfacerse exactamente. En nuestro ejemplo, n 20, por lo que por (w**xj -I- fi*) - 1. Dado que xj - 0 y p - -1, esto significa h* = -1, exactamente como en el Ejemplo 8.2. Por lo tanto, p(x) = signativo. Entonces, p(x) = sign(z - z2 - 1). O

En la práctica, la matriz Qp de *N* x *N* suele ser densa (contiene muchos elementos distintos de cero). Si *N* 100.000, almacenar Qp consume más de 10 GB de RAM. Por lo tanto, para aplicaciones a gran escala, a menudo se utilizan paquetes QP especialmente adaptados que calculan Qp dinámicamente y utilizan propiedades específicas de SVM para resolver (8.22) eficientemente.

8.2.3 Recuperación de la SVM a partir de la solución dual

Ahora que hemos resuelto el problema dual mediante programación cuadrática para obtener la solución óptima n*, lo que queda por hacer es calcular el hyperplano óptimo (h*, w*). Los pesos son fáciles, y se obtienen utilizando la condición estacionaria en (8.19):

$$\mathbf{w}^* = \sum^{N} y_n \alpha_n^* \mathbf{x}_n. \tag{8.23}$$

Supongamos que los datos contienen al menos un ejemplo positivo y otro negativo (de lo contrario, el problema de clasificación es trivial). Entonces, al menos uno de los (o } será estrictamente positivo. Sea o} 0 (s para el vector *soporte*). La condición de holgura complementaria de KKT en el Teorema 8.7 nos dice que la restricción correspondiente a este n{ distinto de cero es la igualdad. Esto significa que

$$y_s \left(\mathbf{w}^{*\mathrm{T}} \mathbf{x}_s + b^* \right) = 1.$$

Podemos resolver b* en la ecuación anterior para obtener

$$\begin{matrix} \text{*- W** X} \\ y_s - \sum^{N} y_n \alpha_n^* \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_s. \end{matrix} \qquad (824)$$

Ejercicio 8.12

Si todos los datos son de una clase, entonces o\0 para n1 , ... , *N.*

(a) ¿Qué es "w"?

(b) ¿Qué es 6"?

Las ecuaciones (8.23) y (8.24) conectan el óptimo n*, que obtenemos de resolver el problema dual, con el hiperplano óptimo (h*, w*) que resuelve (8.4). Y lo que es más importante, la hipótesis óptima es

$$\#(x) = \text{sgn}(w\ "x+/) \left(\begin{array}{l} \bar{w} \\ \quad S1\#H\$pG\ XpXL\# \end{array} \right)' \qquad (8.25)$$

$$\text{sign}\left(\sum^{N} y_n \alpha_n^* \mathbf{x}_n^{\mathrm{T}} (\mathbf{x} - \mathbf{x}_s) + y_s \right)$$

Recordemos que (x, , p,) es un vector soporte que se define por la condición o* 0 . Hay una suma sobre n en las ecuaciones (8.23), (8.24) y (8.25), pero sólo los términos con op 0 contribuyen a las sumas. Por tanto, podemos obtener una representación más eficiente para p(x) utilizando sólo los o positivos :

$$p(x) = \text{signo}\left(9q\ n\ xp\ x + b^* \right) (8.26)$$

donde b* viene dado por (8.24) (el sumatorio (8.24) también puede restringirse a sólo aquellos o 0) . Por lo tanto, p(x) está determinado sólo por aquellos ejemplos (xp, pp) para los que up 0. Resumimos nuestra larga discusión sobre el dual en el siguiente cuadro de algoritmo.

SVM de margen duro con doble QP

i: Construye Qp y Ap como en el Ejercicio 8.11

$$\text{ID} = \begin{bmatrix} \text{fi}\,\text{Vix2Xi} & \cdots & \text{Sr}\,\text{VsVwXsXw} \\ y_N y_1 \mathbf{x}_N^{\mathrm{T}} \mathbf{x}_1 & \cdots & y_N y_N \mathbf{x}_N^{\mathrm{T}} \mathbf{x}_N \end{bmatrix} \quad \text{y Ap} = \begin{bmatrix} \mathbf{y}^{\mathrm{T}} \\ \text{-y"} \\ I_{N \times N} \end{bmatrix}$$

2: Utilizar un solucionador Q P para optimizar el problema dual:

$$\boldsymbol{\alpha}^* \leftarrow \text{QP}(\text{Q}_{\text{D}}, -\mathbf{1}_N, \text{A}_{\text{D}}, \mathbf{0}_{N+2}).$$

3: Sea s un vector soporte para el que o*0 . Calcule b*,

$$\text{°'\_'-Z'.O:}^{\text{X}}\text{.X.}$$

4: Devolver la hipótesis final

$$(\text{X - S1\#H} \left(\quad \text{SpGqXqX-/\#'} \right)$$

Los *sectores de apoyo* son los ejemplos (x" 9,) para los que n{ 0. Los vectores de apoyo desempeñan dos funciones importantes. En primer lugar, sirven para identificar los puntos de datos en el límite del hiperplano graso óptimo. Esto se debe a la condición de holgura complementaria del teorema de KKT:

$$y_s \left(\mathbf{w}^{*\mathrm{T}} \mathbf{x}_s + b^* \right) = 1.$$

Esta condición identifica los vectores de apoyo como los más cercanos a, de hecho, en el límite de la grasa óptima-hiperplano. Esto nos lleva a una interpretación geométrica interesante: la SVM dual identifica los vectores de soporte en el límite del hiperplano de grasa óptimo y utiliza sólo esos vectores de soporte para construir el clasificador final. Ya hemos destacado estos vectores de apoyo en la Sección 8.1, donde utilizamos el término vector de apoyo para destacar el hecho de que estos puntos de datos están "apoyando" el cojín, evitando que se expanda más.

Ejercicio 8.13

La holgura complementaria de KKT da que si o¿ > 0, entonces (xn, //n) está en la frontera del hiperplano gordo óptimo y y (w*"x + b") 1.

Demuestre que lo contrario no es cierto. Es decir, es posible que o\0 y sin embargo (x , y) está en el límite que satisface //n (w""x + b") 1.

{Sugerencia Considere un conjunto de datos de juguete *con dos ejemplos positivos en* (0, 0) *y* (1, 0), *y un ejemplo negativo en* (0, 1).

El ejercicio anterior dice que a partir del dual identificamos un subconjunto de puntos en la frontera del hiperplano graso óptimo que se denominan vectores soporte. Todos los puntos de la frontera son *candidatos a* vectores soporte, pero sólo lo s o n los que tienen rig 0 (y contribuyen a w*).

El segundo papel que desempeñan los vectores de soporte es determinar la hipótesis final p(x) mediante (8.26). El problema dual de la SVM identifica directamente los puntos de datos relevantes para la hipótesis final. Observe que *sólo* se necesitan estos vectores de soporte para calcular p (x).

Ejemplo 8.9. En el Ejemplo 8.8 calculamos n* (y, y, 1, 0) para nuestro problema de juguete. Como n* es positivo, podemos elegir (x , p) - (0, -1) como nuestro vector soporte para calcular h* en (8.24). La hipótesis final es

$$g(x) = \operatorname{sgn}(-j.\ 22).\ /)\mathrm{I}\ 2\ .\ ,j\ -1)$$

signo (zt - +2 - 1) ,

calculado ahora por tercera vez . □

Utilizamos el hecho de que sólo se necesitan los vectores de soporte para calcular la hipótesis final para derivar el límite superior de A dado en (8.8); este

límite superior sólo depende del número de vectores de soporte. En realidad, nuestro límite

en fi se basa en el número de vectores soporte candidatos. Dado que no todos los vectores de soporte candidatos son vectores de soporte, normalmente se obtiene un límite mucho mejor para A utilizando en su lugar el número de vectores de soporte. Es decir,

$$E_{\text{cv}} \leq \frac{\text{number of } \alpha_n^* > 0}{N}. \tag{8.27}$$

Para probar este límite, es necesario demostrar que si se descarta cualquier punto de datos con rig = 0, la hipótesis final no cambia. El siguiente ejercicio le pide que haga exactamente esto.

Ejercicio 8.14

Supongamos que eliminamos un punto de datos (x , y) con o¿ 0 .

(a) Demuestre que la solución óptima anterior a" sigue siendo factible para el nuevo problema dual (8.21) (después de eliminar ctj)

(b) Demuestre que si existe alguna otra solución factible para el nuevo dual que tenga un valor objetivo menor que a", esto contradiría la optimalidad de ct" para el problema dual original.

(c) Por lo tanto, demuestre que ct" (minusa) es óptimo para el nuevo dual.

(d) Por lo tanto, mostrar que el óptimo grasa-hiperplano no cambió.

(e) Demostrar el límite de Ain (8.27).

En la práctica, no suele haber muchos vectores soporte, por lo que el límite de (8.27) puede ser bastante bueno. La figura 8.8 muestra un conjunto de datos con 50 puntos de datos aleatorios y el hiperplano óptimo resultante con 3 vectores de soporte (en recuadros negros). Los vectores soporte se identifican a partir de la solución dual (rig **O**). Figura 8.8

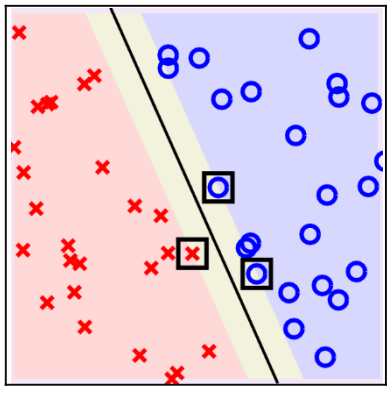


Figura 8.8: Vectores de apoyo identificados a partir de la SVM dual (3 recuadros negros).

apoya el hecho de que a menudo sólo hay unos pocos vectores de apoyo. Esto significa que el n* óptimo suele ser *disperso y* contener muchos elementos cero y unos pocos no ceros. La propiedad de escasez significa que la representación de p(x) en (8.26) puede calcularse eficazmente utilizando sólo estos pocos vectores de soporte. Si hay muchos vectores de soporte, suele ser más eficiente calcular (b*, w*) por adelantado y utilizar p(x) = sign(w**x + b*) para la predicción.

8.3 Truco del núcleo para SVM

Anunciamos el núcleo como una forma de utilizar transformadas no lineales en espacios de alta di- mensión de manera eficiente. Ahora vamos a cumplir esa promesa para la SVM. Para acoplar el kernel con la SVM, necesitamos ver la SVM desde la formulación dual. Y es por eso que hemos invertido un esfuerzo considerable para entender esta visión dual alternativa de SVM. El k e r n e l, junto con la formulación dual, nos permitirá ejecutar eficientemente SVM con transformaciones a espacios de alta o incluso infinita dimensión.

8.3.1 Truco del núcleo mediante SVM dual

Empecemos por revisar el procedimiento para resolver SVM no lineal a partir de la formulación dual basada en una transformación no lineal 4' : I - ' N, lo que puede hacerse sustituyendo x por z &(x) en el cuadro de algoritmo de la página 8-30. Primero, calcule los coeficientes para el problema dual que incluye la matriz Qp; luego resuelva el problema dual para identificar los multiplicadores de Lagrange op distintos de cero y los vectores de soporte correspondientes (xp , pp); finalmente, utilice uno de los vectores de soporte para calcular h*, y devuelva la hipótesis p(x) basada en h*, los vectores de soporte y sus multiplicadores de Lagrange.

En todo el procedimiento, el único paso que puede depender de *d,* que es la dimensión de &(x), es en el cálculo del producto interior del espacio I

$$@\ (Xt\ @\ (X')\ .$$

Este producto interno es necesario en la formulación de Qp y en la expresión para 9(x). El "truco del núcleo" se basa en la siguiente idea: si la transformada *y* el producto interno pueden calcularse conjunta y eficientemente de forma independiente de d, todo el procedimiento SVM no lineal puede llevarse a cabo sin calcular/almacenar cada 4'(x) explícitamente. Entonces, el procedimiento puede funcionar eficientemente para un *d* grande o incluso infinito.

Así que la pregunta es, ¿podemos hacer la transformada y calcular el producto interior de forma eficaz, independientemente de *d?* Definamos primero una función que combine la transformada y el producto interior:

$$K_\Phi(\mathbf{x}, \mathbf{x}') \equiv \Phi(\mathbf{x})^{\mathrm{T}} \Phi(\mathbf{x}'). \qquad (8.28)$$

Esta función se denomina *función kernel,* o simplemente *kernel.* El núcleo toma como entrada dos vectores en el espacio I y da como resultado el

producto interior que

en el espacio I (para la transformada &). En su forma explícita (8.28), parece que el núcleo transforma las entradas x y x' en el espacio y luego calcula el producto interior. La eficiencia de este proceso dependería sin duda de la dimensión del espacio I, que es *d.* La cuestión es si podemos calcular Kg (x, x') de forma más eficiente.

Para dos casos de transformadas no lineales específicas 4', vamos a demostrar que sus correspondientes funciones kernel K¿, pueden calcularse eficientemente, con un coste proporcional a d en lugar de d. (Por simplicidad, usaremos N en lugar de Kg cuando & esté claro por el contexto).

Núcleo polinómico. Consideremos la transformada polinómica de segundo orden:

$$2(\mathrm{X}) \quad - \quad (1, x_1, x_2, \ldots, x_d, x_1x_1, x_1x_2, \ldots, x_dx_d),$$

donde *d* - 1 -I- d -I- d^2 (làs caŕacterísticas idénticas z z, y +y +, se incluyen separadamente por conveniencia matemática como se verá más adelante). Un cálculo directo de &2 (x) requiere *O(d)* de tiempo, por lo que un cálculo directo de &2 (x) requiere *O(d)* de tiempo, por lo que un cálculo directo de &2 (x) requiere *O(d) de* tiempo.

lg(X)' 2(X')

$$x_ix_jx'_ix'_j \sum\sum + x_ix'_i \sum +1$$

también tarda *O(d).* Podemos simplificar la doble suma reorganizando los términos en un producto de dos sumas separadas:

$$\sum_{i=1}^{d}\sum_{j=1}^{d} x_ix_jx'_ix'_j = \sum_{i=1}^{d} x_ix'_i \times \sum_{j=1}^{d} x_jx'_j = (\mathbf{x}^{\mathrm{T}}\mathbf{x}')^2.$$

Por lo tanto, podemos calcular (x)" (x') mediante una función equivalente

K(x x)' 1+(x'x)+x'x)i

En este caso, vemos que N puede calcularse fácilmente en tiempo *O(d),* que es asintóticamente más rápido que *d.* Así, hemos demostrado que, para la transformada polinomial, el núcleo N es un atajo matemático y computacional que nos permite combinar la transformada *y* el producto interior en una única función más eficiente.

Si el kernel N es eficientemente computable para algún & específico, como es el caso de nuestra transformada polinómica, entonces siempre que necesitemos calcular el producto interior de las entradas transformadas en el espacio I podemos utilizar el *kernel brick* y en su lugar calcular la función kernel de esas entradas en el espacio I . Cualquier técnica de aprendizaje que utilice productos internos puede beneficiarse de este truco del kernel. En particular, volvamos al problema dual SVM transformado en el espacio I . Obtenemos el problema dual en el espacio I sustituyendo cada instancia de xp en (8.21) por zp 4'(xp). Después de hacer esto, el

dual

problema de optimización se convierte en:

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{minimizar}} \quad \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \mathbf{z}_n^T\mathbf{z}_m - \sum_{n=1}^{N}\alpha_n \tag{8.29}$$

$$\text{sujeto a:} \quad \sum_{n=1}^{N} y_n\alpha_n = 0$$

$$\alpha_n \geq 0 \qquad (n=1,\ldots,N)$$

Recordemos ahora los pasos para obtener nuestra hipótesis final. Resolvemos el dual para obtener el óptimo n*. Para un vector soporte (z" 9,) con o{ 0 , definimos b* = p, - pq pqop zq z,. Entonces, la hipótesis final de (8.25) es

$$q(x) = \text{signo}\left(\sum_{\text{p Op zp z}} \alpha_n y_n \mathbf{z}_n^T\mathbf{z} - b^*\right),$$

donde z 4'(x). En toda la formulación dual, zq y z sólo aparecen como productos internos. Si utilizamos el truco del núcleo para reemplazar cada producto interno con la función del núcleo, entonces el proceso en/ire de resolver el dual para obtener la hipótesis final será en términos del núcleo. Nunca tenemos que visitar el espacio I para construir explícitamente zq o z. El cuadro de algoritmo a continuación resume estos pasos.

SVM de margen duro con núcleo

i: Construye Qp a partir del núcleo N, y Ap:

$$Q_D \quad \begin{matrix} y_1y_1K_{11} & \ldots & y_1y_NK_{1N} \\ \text{§2? K21} & \text{:.} & \\ \text{§2VyK2W} & & \\ y_Ny_1K_{N1} & \ldots & y_Ny_NK_{NN} \end{matrix} \qquad \text{yAp} = \begin{matrix} \mathbf{y}^T \\ \text{-y*} \\ \text{I} \end{matrix}$$

donde Kpq K(xq , xq). (K se llama la matriz de.) 2: Utilice un Q P-solver para optimizar el problema dual:

$$\boldsymbol{\alpha}^* \leftarrow \text{QP}(Q_D, -\mathbf{1}_N, A_D, \mathbf{0}_{N+2}).$$

3: Sea s cualquier vector soporte para el que o{ 0. Calcule

$$\text{°"-- Z} \quad \text{-°. (X-,X;)}$$

4: Devolver la hipótesis final

$$g(x) = \text{sign}\left(\sum_{\text{o}} \text{o,K(x ,x)} + M\right)$$

En el algoritmo, la dimensión del espacio I ha desaparecido, y el tiempo de ejecución depende de la eficiencia del núcleo, y no de *d*. Para nuestro núcleo polinómico, esto significa que la eficiencia viene determinada por *d*.

Una función kernel eficaz se basa en una transformada *específica* cuidadosamente construida para permitir un cálculo rápido. Si consideramos otra transformada de 2º orden

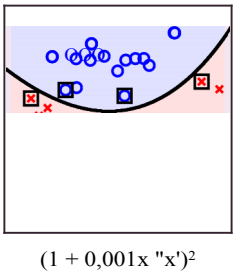
$$@(\text{Xt} \qquad (3, 1 \cdot x_1, 4 \cdot x_2, \ldots, 1 \cdot x_d, 5 \cdot x_1 x_1, 9 \cdot x_1 x_2, \ldots, 2 \cdot x_d x_d)$$

que multiplica cada componente de la transformada original por algún coeficiente arbitrario, sería difícil derivar un núcleo eficiente, aunque nuestros coeficientes estén relacionados con el número mágico. Sin embargo, se pueden tener ciertas combinaciones de coeficientes que harían que K siguiera siendo fácil de calcular. Por ejemplo, fijemos los parámetros 0 y **O** , y consideremos la transformación

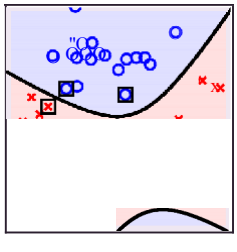
$$\Phi(\mathbf{x}) \qquad \Big(\zeta \cdot 1, \sqrt{2\gamma\zeta} \cdot x_1, \sqrt{2\gamma\zeta} \cdot x_2, \ldots, \sqrt{2\gamma\zeta} \cdot x_d, \gamma \cdot x_1 x_1, \gamma \cdot x_1 x_2, \ldots, \gamma \cdot x_d x_d\Big),$$

entonces N(x, x') (§ + x "x')² , que también es fácil de calcular. El núcleo resultante K suele denominarse núcleo polinómico de segundo orden.

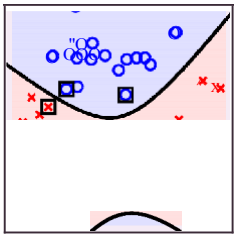
A primera vista, la libertad de elegir (, /) y seguir obteniendo un núcleo eficientemente computable parece útil y también inofensiva. Multiplicar cada característica en el espacio I por diferentes constantes no cambia el poder expresivo de los clasificadores lineales en el espacio I. Sin embargo, cambiar (, §) cambia la peorne/rp en el espacio I, lo que afecta a las distancias y, por tanto, a la rnorpina de un hiperplano. Así, diferentes (, /) podrían dar como resultado un hiperplano óptimo diferente en el espacio N, ya que los márgenes de todos los hiperplanos han cambiado, y esto puede dar un separador cuadrático diferente en el espacio I. Las siguientes figuras muestran lo que ocurre en algunos datos artificiales cuando se varía con fijo a 1.



(1 + 0,001x "x')²



1 -|- (x x') + (x x')²



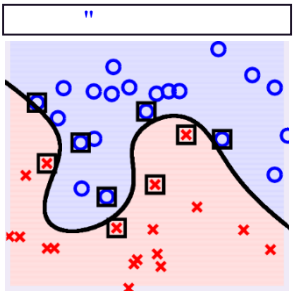
(1 + 1000x "x')²

Vemos que las tres curvas cuadráticas son diferentes, al igual que sus vectores de soporte, indicados por los cuadrados. Es difícil, incluso después de un poco de fisgoneo visual "prohibido", decidir qué curva es mejor. Una posibilidad es utilizar el límite A" basado en el número de vectores de soporte, pero hay que tener en cuenta que el límite A" puede ser bastante impreciso en la práctica. Otras posibilidades

incluyen el uso de las otras herramientas de validación que hemos introducido en el Capítulo 4 para elegir u otros parámetros en la función kernel. La elección de los kernels y de los parámetros del kernel es bastante importante para garantizar un buen rendimiento de la SVM no lineal. En la sección 8.3.2 se discutirán algunas pautas sencillas para los kernels más populares.

La derivación del núcleo *polinómico* de segundo orden anterior puede extenderse al popular *núcleo polinómico de grado-Q*

$$K(\mathbf{x}, \mathbf{x}') = (\zeta + \gamma \mathbf{x}^{\mathrm{T}} \mathbf{x}')^{Q},$$



donde 0 , 0, y Q C H". Entonces, con el truco del kernel, la SVM de margen duro puede aprender límites polinómicos sofisticados de diferentes órdenes utilizando exactamente el mismo procedimiento y simplemente introduciendo diferentes kernels polinómicos. Por lo tanto, podemos utilizar eficientemente
kernels de alta dimensión y, al mismo tiempo, controlar implícitamente la complejidad del modelo maximizando el margen. La figura lateral muestra un polinomio de orden 10 encontrado por kernel-SVM con margen 0,1.

N ú c l e o Gaussiano-RBF. Otro kernel popular es el llamado Gaussian-Núcleo RBF", que tiene la forma

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$

para algún 0. Tomemos = 1 y tomemos x como un escalar z C R para entender la transformada & implicada por el núcleo. En este caso

```
(1,1') - exp - I-I'
       = exp -( 2).exp(2iñ).expr-( )
```

$$\text{exp - (i))}^{2} \left(\frac{2^{k} \; (/ \; {}^{k}\!/\!(\; k}{)} \right)_{k-} \cdot \exp\left(-(x')^2\right),$$

que equivale a un producto interno en un espacio de características definido por la transformada no lineal

$$\text{T'(z)} \qquad \exp(^{-z2}) \left(1, \quad \sqrt{\frac{2^{1}}{1!}} x, \sqrt{\frac{2^{2}}{2!}} x^2, \sqrt{\frac{2^{3}}{3!}} x^3, \quad \right)$$

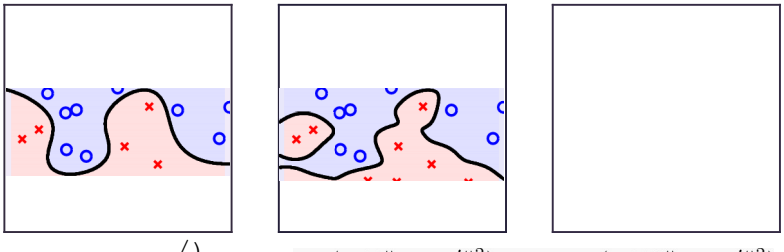
Obtuvimos esta transformada no lineal dividiendo cada término de la serie de Taylor de N(z, z') en términos idénticos que implican z y z'. Nótese que en este caso, & es

Nos ceñimos a la notación Q para el orden de un polinomio, que no debe confundirse con la matriz Q de la programación cuadrática.

"RBF" proviene de la función de base radial introducida en el capítulo fi. Utilizamos el parámetro en lugar del parámetro de escala 1/r, que es común en el contexto de SVM.

una forma trans finile - dimensional. Por lo tanto, un cálculo directo de &(z)*&(z') no es posible en este caso. No obstante, con el truco del kernel, la SVM de margen duro puede encontrar un hiperplano en el espacio N de dimensión infinita con una complejidad del modelo bajo control si el margen es lo suficientemente grande.

El parámetro controla la anchura del n ú c l e o gaussiano. Diferentes opciones para la anchura corresponden a diferentes geometrías en el espacio infinito dimensional I, al igual que diferentes opciones para (, §) en el kernel polinomial corresponden a diferentes geometrías en el espacio polinomialI. Las siguientes figuras muestran los resultados de clasificación de tres núcleos RBF gaussianos que sólo difieren en la elección de .



Cuando las funciones gaussianas son estrechas (grandes), vemos claramente que ni siquiera la protección de un margen grande puede suprimir el sobreajuste. Sin embargo, para un > razonablemente pequeño, el sofisticado límite descubierto por SVM con el kernel Gaussiano-RBF parece bastante bueno. Una vez más, esto demuestra que los núcleos y los parámetros de los núcleos deben elegirse cuidadosamente para obtener un rendimiento razonable.

Ejercicio 8.15

Consideremos dos transformadas de (eatura de dimensión finita 'I'1 e 'I'2 y sus correspondientes kernels Ai y 6z

(a) Defina 6(x) (I'1(X), 'I'2 (x)). Expresar el núcleo correspondiente de 'T' en términos de Ai y 6z.

(b) Consideremos la matriz "I'1 (x) I'2 (x)" y sea 6(x) la representación vectorial de la matriz (por ejemplo, concatenando todas las filas). Exprese el núcleo correspondiente de 6 en términos de R1 y kz.

(c) Por lo tanto, demuestre que si Ni y 6z son núcleos, entonces también lo son $K_1 K_2$.
Ni + 6z y

Los resultados anteriores pueden utilizarse para construir los núcleos polinómicos generales y (cuando se extienden a las transformaciones de

dimensión infinita) para construir los núcleos gaussianos-RBF generales.

8.3.2 Elección de los granos

En la práctica se utilizan tres núcleos: lineal, polinómico y gaussiano-RBF. El k e r n e l lineal, que corresponde a un kernel polinomial especial con Q = 1, 1 , y = 0, corresponde a la transformada de identidad. Resolver el problema dual SVM (8.22) con el kernel lineal es equivalente a resolver el SVM lineal de margen duro (8.4). Muchos paquetes especiales de SVM utilizan la equivalencia para encontrar el óptimo (b*, w*) o n* más eficientemente. Una ventaja particular de la SVM lineal de margen duro es que el valor de v* puede llevar alguna explicación sobre cómo se hace la predicción p(x), al igual que nuestro perceptrón familiar. Una desventaja particular de la SVM lineal de margen duro es la incapacidad de producir un límite sofisticado, que puede ser necesario si los datos no son linealmente separables.

El núcleo polinómico proporciona dos controles de complejidad: un control explícito del grado Q y un control implícito del concepto de margen grande. El núcleo puede funcionar bien con una elección adecuada de (, §) y Q. Sin embargo, elegir una buena combinación de tres parámetros no es tarea fácil. Además, cuando Q es grande, el núcleo polinómico se evalúa a un valor con una magnitud muy grande o muy pequeña. El amplio rango de valores introduce dificultades numéricas a la hora de resolver el problema dual. Por lo tanto, el núcleo polinómico se utiliza normalmente sólo con grado Q 10, e incluso entonces, sólo cuando

+ wx*x' se escalan a valores razonables eligiendo adecuadamente (§, w).

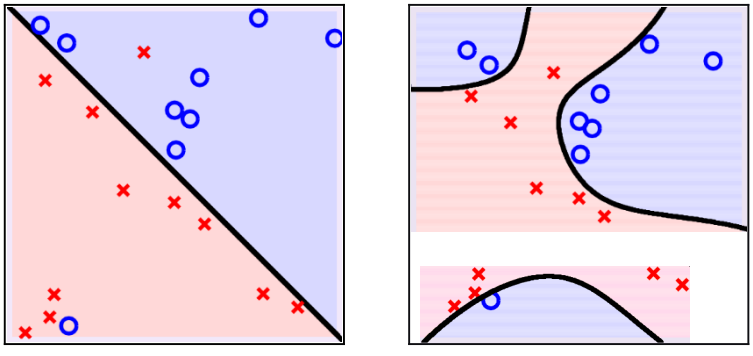
El kernel Gaussiano-RBF puede conducir a un límite sofisticado para SVM mientras se controla la complejidad del modelo utilizando el concepto de margen grande. Sólo es necesario especificar un parámetro, la anchura , y su rango numérico se elige universalmente en el intervalo 0, 1) . Esto hace que a menudo sea preferible a los núcleos polinómicos y lineales. En el lado negativo, como la transformación correspondiente del núcleo gaussiano-RBF es de dimensión infinita, la hipótesis resultante sólo puede expresarse mediante los vectores de soporte en lugar del hiperplano real, lo que dificulta la interpretación de la predicción q(x). Curiosamente, cuando se combina la SVM con el kernel gaussiano-RBF, la hipótesis contiene una combinación lineal de funciones gaussianas centradas en xp , que puede considerarse como un caso especial de la red RBF introducida en el e-Capítulo 6.

Además de los tres núcleos anteriores, existen muchas otras opciones. Incluso se pueden diseñar nuevos núcleos que se adapten mejor a la tarea de aprendizaje en cuestión. Tenga en cuenta, sin embargo, que el núcleo se define como una función abreviada para calcular productos internos. Así, de forma similar a lo que se muestra en el Ejercicio 8.11, la matriz from definida por

$$\mathrm{K} \quad \begin{matrix} K(\mathbf{x}_1,\mathbf{x}_1) & K(\mathbf{x}_1,\mathbf{x}_2) & \dots & K(\mathbf{x}_1,\mathbf{x}_N) \\ K(\mathbf{x}_2,\mathbf{x}_1) & K(\mathbf{x}_2,\mathbf{x}_2) & \dots & K(\mathbf{x}_2,\mathbf{x}_N) \\ \\ K(\mathbf{x}_N,\mathbf{x}_1) & K(\mathbf{x}_N,\mathbf{x}_2) & \dots & K(\mathbf{x}_N,\mathbf{x}_N) \end{matrix}$$

debe ser semidefinida positiva. Resulta que esta condición no sólo es necesaria

sino también suficiente para que N sea una función de núcleo válida que corresponda a



(a) clasificador lineal (b) Núcleo gaussiano-RBF

Figura 8.9: Para (a) un conjunto de datos ruidosos, el clasificador lineal parece funcionar bastante bien, (b) el uso del kernel gaussiano-RBF con la SVM de margen duro conduce a un sobreajuste.

al producto interior en algún espacio no lineal I no lineal. Este requisito se conoce formalmente como condición de Mercer.

> K(x, x') es una función de núcleo válida si y sólo si la matriz kernel K es siempre PSD simétrica para cualquier (x xy} dada.

Esta condición se utiliza a menudo para descartar funciones de núcleo no válidas. Por otro lado, demostrar que una función del núcleo es válida no es una tarea trivial, incluso cuando se utiliza la condición de Mercer.

8.4 SVM de margen suave

La SVM de margen duro asume que los datos son separables en el espacio I

.

Cuando transformamos x a un espacio I de alta dimensión, o a uno de dimensión infinita utilizando, por ejemplo, el kernel Gaussiano-RBF, podemos fácilmente sobreajustar los datos. La figura 8.9(b) muestra esta situación. La SVM de margen duro combinada con el kernel gaussiano-RBF insiste en ajustar los dos "valores atípicos" mal clasificados por el clasificador lineal, y da como resultado un límite de decisión innecesariamente complicado que debería hacer sonar su "alarma de sobreajuste".

Para los datos de la Figura 8.9(b), deberíamos utilizar un hiperplano lineal simple en lugar del complejo separador Gaussiano-RBF. Esto significa que tendremos que aceptar algunos errores, y la SVM de margen duro no puede acomodar

7Se puede demostrar que el kernel gaussiano-RBF puede separar cualquier conjunto de datos (sin
puntos repetidos xp).

ya que está diseñado para la clasificación perfecta de datos separables. Un remedio es considerar una formulación "suave": tratar de obtener un hiperplano de márgenes grandes, pero permitir pequeñas violaciones de los márgenes o incluso algunos errores de clasificación. Como veremos, esta formulación controla la sensibilidad del modelo SVM a los valores atípicos.

La formulación más común de la *SVM* (lineal) so -rnorpin es la siguiente. Introducir una "cantidad" de violación del margen 5q 0 para cada punto de datos (xp, pp) y exigir que

$$y_n \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b\right) \geq 1 - \xi_n.$$

Según nuestra definición de separación en (8.2), (xp , pp) se separa si pp (w "xp -I- fi) **1** , por lo que (p captura en qué medida (xp , pp) no se separa. En términos de margen, recuerde que si 9q (w*xp -I- b) = 1 en la SVM de margen duro, entonces el punto de datos está en el límite del margen. Por lo tanto, (p cap- tura hasta dónde puede llegar el punto de datos en el margen. El margen ya no es duro, es "blando", lo que permite que un punto de datos penetre en él. Obsérvese que si un punto (xp , pp) satisface pq (w*xp -I- fi) 1 , entonces no se viola el margen y la f;q correspondiente se define como cero. Idealmente, nos gustaría que la suma total de las violaciones del margen fuera pequeña, por lo que modificamos la SVM de margen duro a la SVM de margen suave permitiendo las violaciones del margen pero añadiendo un término de penalización para desalentar las violaciones grandes. El resultado es el problema de optimización de margen suave:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n \tag{8.30}$$

sujeto topp (w "xp -I- fi) 1 " para n - 1, 2, ... , *N,*

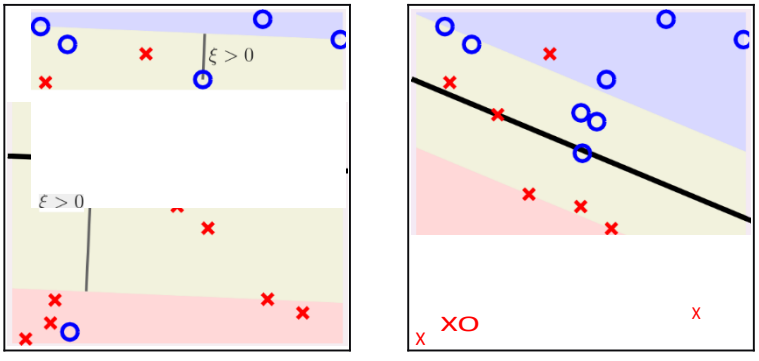
" *II* para n = 1, 2, ... , N.

Resolvemos este problema de optimización para obtener (b, w). En comparación con (8.4), los nuevos términos aparecen resaltados en rojo. Obtenemos un gran margen minimizando el término $w*w en la función objetivo. Obtenemos pequeños incumplimientos minimizando

el término (p. Al minimizar la suma de estos dos términos, obtenemos un compromiso entre nuestros dos objetivos, y ese compromiso favorecerá a uno u otro de nuestros objetivos (violaciones de margen grande frente a violaciones de margen pequeño) dependiendo del parámetro de penalización definido por el usuario denotado por U. Cuando U es grande, significa que nos preocupamos más por violar el margen, lo que nos acerca a la SVM de margen duro. Cuando C es pequeño, por otro lado, nos preocupamos menos por violar el margen. Al elegir C adecuadamente, obtenemos un hiperplano de margen grande (dpp efectivo pequeño) con una pequeña cantidad de violaciones de margen.

El nuevo problema de optimización en (8.30) parece más complejo que (8.4). No se asuste. Podemos resolver este problema utilizando programación cuadrática, igual que hicimos con la SVM de margen duro (véase el Ejercicio 8.16 para la solución explícita). La Figura 8.10 muestra el resultado de resolver

(8.30) utilizando los datos de la Figura 8.6(a). Cuando U es pequeño, obtenemos un clasificador con un margen muy grande pero con muchos casos de violación del margen (algunos cruzan el límite y otros no); cuando



(a) = 1 (b) U = 500

Figura 8.10: El algoritmo lineal de hiperplano óptimo de margen suave del Ejercicio 8.16 sobre los datos no separables de la Figura 8.6(a).

U es grande, obtenemos un clasificador con menor margen pero con menor violación del margen.

Ejercicio 8.16

Demuestre que el problema de optimización en (8.30) es un problema QP.

(a) Demuestre que la variable de optimización es $u = \begin{bmatrix} b \\ w \\ \xi_1 \\ \vdots \\ \xi_N \end{bmatrix}$, donde g

(b) Demuestre que u*QP (@, p, A, c), donde

$$Q = \begin{bmatrix} 0 & \overset{d}{0_d} & \overset{N}{0} \\ 0_d & I_d & 0_{d\times N} \\ 0_N & 0_{N\times d} & 0_{N\times N} \end{bmatrix}, \quad p = \begin{bmatrix} 0_{d+1} \\ C\cdot 1_N \end{bmatrix}, \quad A = \begin{bmatrix} YX & I \\ 0_N & I_N \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0_p \end{bmatrix},$$

e YX es la matriz de datos con signo del Ejercicio 8.4.

(c) ¿Cómo se recuperan b", w" y g" a partir de u"?

(d) ¿Cómo se determina qué puntos de datos violan el margen, qué puntos de datos están en el borde del margen y qué puntos de datos están correctamente separados y fuera del margen?

Al igual que el problema de optimización de margen duro (8.4), la versión de margen blando (8.30) es un problema Q P convexo. El problema dual correspondiente puede derivarse utilizando la misma técnica introducida en la sección 8.2. Aquí nos limitaremos a mostrar el primer paso y dejaremos que usted termine el resto. Para el margen suave

SVM (8.30), la función de Lagrange es

$$\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\alpha_n(1-\xi_n - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)) - \sum_{n=1}^{N}\beta_n\xi_n,$$

donde son los multiplicadores de Lagrange para pp (w "xp -I- fi) 1 - (p , y fig0 son los multiplicadores de Lagrange para;q Entonces, la condición KKT nos dice que en la solución óptima, fi£/fif;q tiene que ser 0. Esto significa que

$$C - \alpha_n - \beta_n = 0.$$

Es decir, op y fig suman U. Podemos entonces sustituir todo fippor -up sin pérdida de optimalidad. Si lo hacemos, el problema dual de Lagrange se simplifica a

$$\max_{\substack{\boldsymbol{\alpha}\geq\mathbf{0},\boldsymbol{\beta}\geq\mathbf{0},\\ \alpha_n+\beta_n=C}} \quad \min_{b,\mathbf{w},\boldsymbol{\xi}} \mathcal{L}(b,\mathbf{w},\boldsymbol{\xi},\boldsymbol{\alpha}),$$

donde

$$\begin{aligned}\mathcal{L} &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\alpha_n(1-\xi_n - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)) - \sum_{n=1}^{N}(C-\alpha_n)\xi_n \\ &= \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N}\alpha_n(1 - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)).\end{aligned}$$

¿Le resulta familiar la función objetivo simplificada? Es justo la función objetivo que nos llevó al dual de la SVM de margen duro. Confiamos en que pueda seguir todos los pasos de la Sección 8.2 para obtener el dual completo. Cuando se expresa en forma matricial, el problema dual de (8.30) es

$$\begin{aligned}\min \quad & \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{Q}_D\boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{sujeto a} \quad & \mathbf{y}^{\mathrm{T}}\boldsymbol{\alpha} = 0;\end{aligned}$$

Curiosamente, en comparación con (8.22), el único cambio del problema dual es que cada aparejo está ahora acotado por C, la tasa de penalización, en lugar de oo. La formulación puede resolverse de nuevo mediante algunos paquetes de programación cuadrática generales o adaptados específicamente.

Para la SVM de margen suave, obtenemos una expresión similar para la hipótesis final en términos de la solución dual óptima, como la de la SVM de margen duro.

$$\#(x) = \sin\left(\frac{pn}{x}\right) x + M$$

El truco del núcleo sigue siendo aplicable siempre que podamos calcular el b* óptimo de forma eficiente. Obtener el óptimo fi* a partir del óptimo n*, sin embargo, es algo más complicado en el caso de margen blando. Las condiciones KKT establecen que

$$\alpha_n^* \cdot \left(y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n + b^*) - 1 + \xi_n^* \right) \quad 0.$$
$$\beta_n^* \cdot \xi_n^* = (C - \alpha_n^*) \cdot \xi_n^* \quad 0.$$

Si n 0 , entonces $pp (w**xp -I- b*) - 1 -I- = 0 y por tanto

$$y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n + b^*) = 1 - \xi_n^* \leq 1.$$

Por otra parte, si up U , entonces (J = 0 y por lo tanto

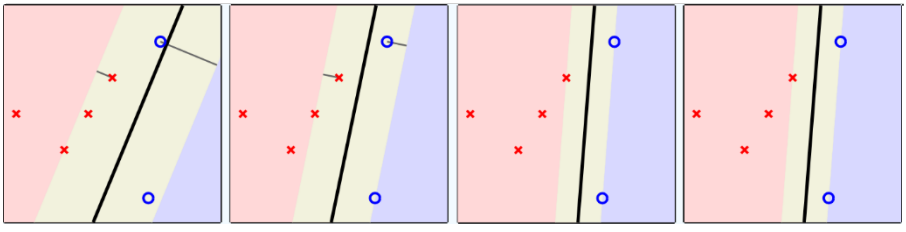
$$y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n + b^*) \geq 1.$$

Las dos desigualdades dan un rango para el óptimo h*. Cuando hay un vector de soporte con 0 arriba U , vemos que las desigualdades se pueden combinar a una igualdad pp(w**xp -I- fi*) - 1, que se puede utilizar para fijar el óptimo h* como lo hicimos para el hard-margin SVM. En otros casos, hay muchas opciones de b* y se puede elegir libremente cualquiera.

Los vectores soporte con 0 o U se denominan v e c t o r e s s o p o r t e libres, que se garantiza que están en el límite del hiperplano graso y, por tanto, también se denominan vectores soporte de margen. En cambio, los vectores soporte con op = C se denominan vectores soporte acotados (también llamados vectores soporte sin margen). Pueden estar en el límite gordo, violar ligeramente el límite pero aún así predecir correctamente, o violar gravemente el límite y predecir erróneamente.

Para datos separables (en el espacio transformado &), existe algún C' tal que siempre que C C', la SVM de margen suave produce exactamente la misma solución que la SVM de margen duro. Por lo tanto, la SVM de margen duro puede verse como un caso especial de la de margen blando, ilustrada a continuación.

Margen blando			**Hard-margn**
peque ño	U mediana de	gran	



Echemos un vistazo más de cerca al parámetro C en la fórmula SVM de margen suave (8.30). Si U es grande, significa que queremos que todas las violaciones (errores) {p sean lo más pequeñas posible, con la posible contrapartida de un margen más pequeño (mayor complejidad). Si U es pequeño, toleraremos una cierta cantidad de errores, mientras que posiblemente obtendremos una hipótesis menos complicada con un gran margen. ¿Le resulta familiar este dilema? En el capítulo 4, cuando estudiamos la regularización, nos encontramos con una solución de compromiso de este tipo. Sea

$$E_{\text{SVM}}(b, \mathbf{w}) = \frac{1}{NZ} \overset{N}{} \max(1- q(w'xpd- ,0).$$

El término n-ésimo en A,pp (h, w) se evalúa a 0 si no hay violación del ejemplo n-ésimo, por lo que pq (w*xp -I- fi) **1** ; de lo contrario, el término n-ésimo es la cantidad de violación para el punto de datos correspondiente. Por lo tanto, el objetivo que minimizamos en SVM de margen suave (8.30) se puede reescribir como el siguiente problema de optimización

min Jw*w -I- A,pp (fi, w),

sujeto a las restricciones, y donde A - 1/2UN. En otras palabras, la SVM de margen suave puede considerarse un caso especial de clasificación regularizada con A pq (b, w) como sustituto del error en la muestra e $_{yw}$ "w (sin b) como regularizador. El término A pq (b, w) es un límite superior del error de clasificación en la muestra A¡p, mientras que el término regularizador procede del concepto de margen grande y controla la complejidad efectiva del modelo.

Ejercicio 8.17

Demuestre que A vu(6, w) es un límite superior en el *T:; (b, w),* donde *T:;* es el error de clasificación 0/1.

En resumen, la SVM de margen suave puede:

1. Entregar un hiperplano de gran margen, y al hacerlo puede controlar la complejidad efectiva del modelo.
2. Trate con transformaciones de alta o infinita dimensión utilizando el truco del núcleo,
3. Expresar la hipótesis final p(x) utilizando sólo unos pocos vectores de soporte, sus correspondientes multiplicadores de Lagrange y el kernel.
4. Control the sensitivity to outliers and regularize the solution through

Cuando el parámetro de regularización U y el kernel se eligen adecuadamente, se observa a menudo que la SVM de margen suave disfruta de un fiqpt bajo con las propiedades útiles anteriores. Estas propiedades hacen que la SVM de margen suave (tfie SVM para abreviar) sea uno de los modelos de clasificación más útiles y, a menudo, la primera opción en el aprendizaje a partir de datos. Es

un modelo lineal robusto con capacidad de transformación no lineal avanzada cuando se utiliza con un kernel.

8.5 Problemas

Problema 8.1 Considere un conjunto de datos con dos puntos de datos xy E @' de clase A1 respectivamente. Resuelva manualmente (8.4) minimizando explícitamente ||w||^2 sujeto a las dos restricciones de separación.

Calcula el hiperplano óptimo (margen máximo) (6", w") y su margen. Compáralo con tu solución del Ejercicio 8.1.

Problema 8.2 Considere un conjunto de datos con tres puntos de datos en @^{2}

$$X = \begin{matrix} 0 & 0 \\ 0 & -1 \\ -2 & 0 \end{matrix} \qquad y = \begin{matrix} -1 \\ -1 \\ 11 \end{matrix}$$

Resuelva manualmente (8.4) para obtener el hiperplano óptimo (b", w") y su margen.

Problema 8.3 Resuelva manualmente la optimización dual del Ejemplo 8.8 para obtener el mismo a" que se obtuvo en el texto utilizando un solucionador QP. Utilice los siguientes pasos.

(a) Demuestre que el problema de optimización dual consiste en minimizar

$$\mathcal{L}(\boldsymbol{\alpha}) = 4\alpha_2^2 + 2\alpha_3^2 + \tfrac{0}{2}\alpha_4 - 4\alpha_2\alpha_3 - 6\alpha_2\alpha_4 + 6\alpha_3\alpha_4 - \alpha_1 - \alpha_2 - \alpha_3 - \alpha_4,$$

sujeta a las restricciones

$$\begin{matrix} *1 + *2 \ ' *3 \quad *4; \\ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0. \end{matrix}$$

(b) Utilice la restricción de igualdad para sustituira en £(a) por

$$\mathcal{L}(\boldsymbol{\alpha}) = 4\alpha_2^2 + 2\alpha_3^2 + \tfrac{0}{2}\alpha_4 \text{ 4O2 3 - 602 4 -! } \quad \text{-6ag4 - 2as - 204 -}$$

(c) Fix3 , o4 0 y minimizar C(a) en (b) con respecto a oz para demostrar que

$$\alpha_2 = \frac{\alpha_3}{2} + \frac{3\,4}{4} \qquad \text{and} \qquad \alpha_1 = \alpha_3 + \alpha_4 - \alpha_2 = \frac{\alpha_3}{2} + \frac{*4}{4}$$

¿Son estas soluciones válidas paraa, a2?

(d) Utilice las expresiones de (c) para reducir el problema a la minimización de

$$\mathcal{L}(\boldsymbol{\alpha}) = \text{nd} + 4\,'4^{\frac{2}{9}} \quad {}^2 \quad \text{3n3} \quad 4 \; - 2\alpha_3 - 2\alpha_4,$$

sujeto a oa, o4 0. Demostrar que el mínimo se alcanza cuando as 1 y a4 0. ¿Qué son o1, o2?

Es un alivio disponer de solucionadores Q P (¡o resolver problemas de este tipo en el caso general!

Problema 8.4 Plantee el problema dual para el conjunto de datos de juguete del Ejercicio 8.2. A continuación, resuelva el problema dual y calcule a", los multiplicadores de Lagrange óptimos. A continuación, resuelva el problema dual y calcule a", los multiplicadores de Lagrange óptimos.

Problema 8.5 [Sesgo y varianza del hiperplano óptimo] En este problema, debe investigar el sesgo y la varianza del hiperplano óptimo en un entorno sencillo. La entrada es (z , z2) E -1, 1]^2 y la función objetivo es /(x) sign(z2).

El conjunto de hipótesisH contiene separadores lineales horizontales /r(x) sign(zz - a), donde -1 a 1 . Consideremos dos algoritmos:

Aleatorio: Elige un separador aleatorio de /Y.

SvM: Elige el separador de margen máximo de /Y.

(a) Genere 3 puntos de datos uniformemente en la mitad superior del espacio de entrada y 3 puntos de datos en la mitad inferior, y obtenga sR. dam y ssVM

(b) Crea un gráfico con tus datos y tus dos hipótesis.

(c) Repita la parte (a) para un millón de conjuntos de datos para obtener un millón de hipótesis Aleatorias y SVM.

(d) Proporcione un histograma de los valores de Random resultantes del algoritmo aleatorio y otro histograma de oSvM resultantes de los separadores óptimos. Compare los dos histogramas y explique las diferencias.

(e) Estime el sesgo y el var de los dos algoritmos. Explica tus conclusiones, en particular qué algoritmo es mejor para este problema de juguete.

Problema 8.6 Demuestre que $\|x - p\|^2$ se minimiza en /z ^{N} x".

Problema 8. 7Para cualquier *x* , . , *xN* con | / x " ||A y *N* par, demuestre que existe una dicotomía equilibrada v1, ... , gn que satisface

$$\sum_{n=1}^{N} y_n = 0, \text{ and } \left\| \sum_{n=1}^{N} y_n \mathbf{x}_n \right\| \leq \frac{NR}{\sqrt{N-1}}.$$

(Este es el lema geométrico que se necesita para acotar la dimensión VC de los hiperplanos p-fat por A /p^{22} -|- 1.) Los siguientes pasos son una guía para la demostración.

Supongamos que seleccionamos al azar N/2 de las etiquetas gi, . , *IN* para que sean 41, siendo las demás -1. Por construcción, *g*< -- 0.

(a) Show $\left\| \sum_{n=1}^{N} y_n \mathbf{x}_n \right\|^2 = \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m.$

(b) Cuando u m, ¿qué es // y, ? Demuestre que @ y "y, I] = $\frac{N}{2}$ - I) /N - 1) cuando u m. Por lo tanto, demuestre que

$$\mathbb{E}[y_n y_m] = \begin{cases} \\ -\frac{1}{N-1} \end{cases} \quad m \neq n.$$

(c) Demuestre que

$$\mathbb{E}\left[\left\|\sum_{n=1}^{N} y_n \mathbf{x}_n\right\|^2\right] \qquad \frac{NH}{-1} \sum_{n=1}^{N} \|\mathbf{x}_n - \bar{\mathbf{x}}\|^2,$$

donde el vector medio x $_{-1}$ #*Hint: Utilice la linealidad de la expec- tación en (a), y considere los casos m -- n y m / n por separado).*

(d) Demuestre ^{N} queZ ^{N} ll-n - x| |^2 | x" | N//^{22} *|Pista: Problema 8.6.)*

(e) Concluir que

$$\mathbb{E}\left[\left\|\sum_{n=1}^{N} y_n \mathbf{x}_n\right\|^2\right] \leq \frac{N^2 R^2}{N-1},$$

y, por tanto, que

$$\mathbb{P}\left[\left\|\sum_{n=1}^{N} y_n \mathbf{x}_n\right\| \leq \frac{NR}{\sqrt{N-1}}\right] > 0.$$

¡Esto significa que para alguna elección de //u,)) gnxn)) < *NR! N* -

Esta prueba se denomina prueba de existencia probabilística: si algún proceso aleatorio puede generar un objeto con *probabilidad positiva,* entonces ese objeto debe existir. Obsérvese que se prueba la existencia de la dicotomía requerida sin construirla realmente. En este caso, la forma más fácil de construir una dicotomía deseada es generar aleatoriamente las dicotomías equilibradas hasta obtener una que funcione.

Problema 8.8 Demostramos que si *N* puntos de la bola de radio A están destrozados por hiperplanos con margen *p,* entonces *N* A /p^{22} -|- 1 cuandoN es par. Consideremos ahora *N impar*, y x , , "*N* con | | x || A destrozados por hiperplanos con margen p. Recordemos que (w, 6) implementa gj , … , // con margen *p* si

$$\rho\|\mathbf{w}\| \leq y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b), \qquad \text{for } n = 1, \ldots, N. \quad (8.31)$$

Demuestre que para *N* -- 2k4 1 (impar), *N* A /p 4^{22} $\frac{1}{N}$ -{- 1 como (ollows:

Consideremos etiquetados aleatorios gj , … , //de los *N* puntos en los que *k* o(las etiquetas son 41 y k4 1 son -1. Defina éu - $\frac{1}{k}$if $y_n = +1$ and $\ell_n =$ if $y_{n_i} = -1$.

(a) Para cualquier etiquetado con *k* etiquetas siendo +1, demuestre, sumando (8.31) y

utilizando la desigualdad de Cauchy-Schwarz, que

$$2\rho \leq \Big\| \sum_{\text{n= 1}}^{N} \ell_n y_n \mathbf{x}_n$$

(b) Demuestre que existe un etiquetado, siendo *k* etiquetas 41, para el cual

$$\sum_{\text{n= 1}}^{N} \ell_n y_n \mathbf{x}_n \Big\| \leq \frac{\text{2NA}}{(N\text{ - 1})\,NJ}$$

(i) Show $\Big\| \sum_{n=1}^{N} \ell_n y_n \mathbf{x}_n \Big\|^2 = \sum_{n=1}^{N} \sum_{m=1}^{N} \ell_n \ell_m y_n y_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m.$

(ii) For $m = n$, show $\mathbb{E}[\ell_n \ell_m y_n y_m] = \dfrac{1}{k(k+1)}$

(iii) For $m \neq n$, show $\mathbb{E}[\ell_n \ell_m y_n y_m] = -\dfrac{}{(N-1)k(k+1)}$.

[Hint: $\mathbb{P}[\ell_n \ell_m y_n y_m = 1/k^2] = k(k-1)/N(N-1).$*]*

(iv) Mostrar $\text{E/n I}^{N}_{\ n} y_n \mathbf{x}_n \Big\|^2 \Big| = \dfrac{N}{(N-1)k(k+1)} \sum_{n=1}^{N} \|\mathbf{x}_n - \bar{\mathbf{x}}\|^2$

(v) Utilice el problema 8 . 6 para concluir la prueba con el problema 8.7.

(c) Utiliza (a) y (b) para demostrar que $N \dfrac{R^2}{\text{p2}} + N \text{"}'$

Problema 8.9 Demostrar que para el caso separable, si se elimina un punto de datos que no es un vector soporte, entonces el clasificador de margen máximo no cambia. Puede utilizar los siguientes pasos como guía. Sea g el clasificador de margen máximo para todos los datos, y g el clasificador de margen máximo después de eliminar un punto de datos que no es un v e c t o r s o p o r t e .

(a) Demuestre que g es un separador para 0, los datos menos el vector no soporte.

(b) Demuestre que el clasificador de margen máximo es único.

(c) Demostrar que si p tiene mayor margen que p en 0, entonces también tiene mayor margen en @, una contradicción. Por lo tanto, concluir que p es el separador de máximo margen (o O.

Problema 8.10 Un vector soporte esencial es aquel cuya eliminación del conjunto de datos cambia el separador de margen máximo. Para el caso separable, demuestre que hay a lo sumo *d* -/- 1 vectores soporte esenciales. Por lo tanto, demuestre que para el caso separable,

$$E_{\text{cv}} \leq \frac{d+1}{N}$$

Problema 8.11 Considere la versión del APA que utiliza el punto de datos mal clasificado xn con menor índice n para la actualización del peso. Supongamos que los datos son separables y se dan en algún orden fijo pero arbitrario, y cuando se elimina un punto de datos, no se altera este orden. En este problema, demuestre que

$$\text{A',(PLA)} \qquad \frac{R^2}{N\,p2\,,}$$

donde *p* es el margen (la mitad de la anchura) del hiperplano de separación de margen máximo que devolvería (por ejemplo) la SVM. Los pasos siguientes son una guía para la prueba.

(a) Utilice el resultado del problema 1.3 para demostrar que el número de actualizaciones que que hace el APA es como máximo

$$T \leq \frac{\overline{\ }^{2}}{\rho^2}$$

(b) Argumentar que esto significa que PLA sólo "visita" como máximo A /jo[22] puntos diferentes en el transcurso de sus iteraciones.

(c) Argumentar que después de dejar fuera cualquier punto (xn, gu) que no sea 'visitado', PLA devolverá el mismo clasificador.

(d) ¿Cuál es el error de exclusión eu (o estos puntos que no se visitaron? Por lo tanto, demostrar el límite deseado en Ar,(PLA).

Problema 8.12 Demuestre que la solución óptima para el hy- perplano óptimo de margen suave (resolviendo el problema de optimización (8.30)) con -+ co será la misma solución que se desarrolló usando programación lineal en el Problema 3.6(c).

Problema 8.13 Los datos (o Figura 8.6(b) se dan a continuación:

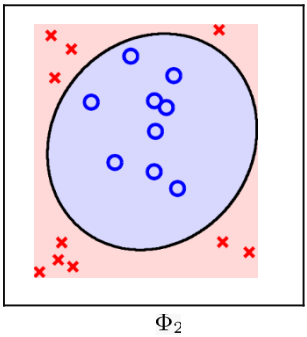
$y_n = +1$	$y_n = -1$
(-0494,0363)	**(0491.0920)**
(-0311,-0101)	**(-0802,-0946)**
(-00064,0£74)	(-0721.-0710)
(-0,0089, -0,tT3)	**(0519, -0715)**
(0UU14,U138)	(-0,77s. 0 s l)
(-0.189, 0.718)	**(-U64%0I3)**
(0.085,0.32208)	(-08030.878)
(0.171, -0.302)	(0.U44,0801)
(0.142,U.566)	(0.724,-0.705)
	(-U.74S. -0853)
	(-0.6X.-0905)

Utilice los datos de la izquierda con las transformadas polinómicas de 2º y 3er orden 2, 'l y el algoritmo pseudoinverso (o regresión lineal del Capítulo 3) para obtener los pesos w de su hipótesis final en el espacio Z. La hipótesis hipótesis en el espacio J es:

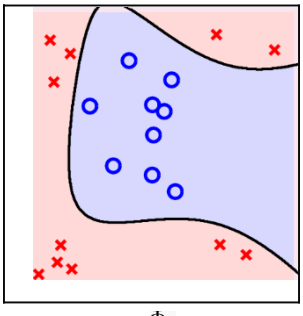
```
g(x) = sign(w'l(x) #).
```

(a) Trace las regiones de clasificación para su hipótesis final en el espacio J.

Los resultados deberían ser parecidos:



Φ_2



Φ_3

(b) ¿Cuál de los ajustes de la parte (a) parece haber sobreajustado?

(c) Utilizar el algoritmo pseudoinverso con el parámetro de regularización 1 para solucionar el sobreajuste que identificó en la parte (c). Represente gráficamente el clasificador resultante.

Problema 8.14 El truco del núcleo se puede utilizar con cualquier modelo siempre que el ajuste de los datos y la hipótesis final sólo requieran el cálculo de productos de puntos en el espacio S. Supongamos que tenemos un kernel C, por lo que

$$\Phi(\mathbf{x})^{\mathrm{T}}\Phi(\mathbf{x}') = K(\mathbf{x}, \mathbf{x}').$$

Sea Z los datos en el espacio J. El algoritmo pseudoinverso para la regresión regularizada calcula los pesos óptimos w* (en el e s p a c i o S) que minimizan

$$E_{\mathrm{aug}}(\tilde{\mathbf{w}}) = \|\mathrm{Z}\tilde{\mathbf{w}} - \mathbf{y}\|^2 + \lambda\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{w}}.$$

La hipótesis final es p(x) sign(w "6(x)). Utilizando el teorema del representador, la solución óptima puede escribirse w" - b*zn Z "/d*

(a) Demuestre que /d" minimiza

$$\mathrm{t}\ )=|\mathrm{Ko-y/^A0'Kd}$$

dondeK es la matriz Kernel-Gram de *N x N* con entradasK C(x xj).

(b) Demuestre queK es simétrico.

(c) Demuestre que la solución al problema de minimización de la parte (a) es:

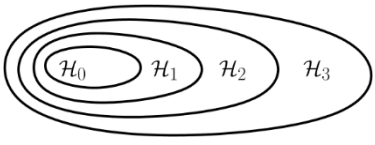
$$\#' = (\mathrm{K}\ \ \mathrm{AI})\ 'y.$$

¿Puede calcularse /d* sin "visitar" nunca el espacio J?

(d) Demuestre que la hipótesis final es

$$g(x) - \mathrm{sign0*N}\ \ (x\ ,x)\ \Big)$$

Problema 8.15 Minimización **del riesgo estructural** (SRM). SRM es un marco de uso(uI para la selección de modelos. Una *estructura* es una secuencia anidada de conjuntos de hipótesis:



Supongamos que utilizamos la minimización del error en la muestra como algoritmo de aprendizaje en cada H , por lo que g argmin Aiu(/r) , y seleccionamos g" argmin *E, g)* 4 U(H) .

(a) Demuestre que el error en la muestra A, (p",) no aumenta con m. ¿Y la penalización U(H",)? ¿Cómo espera que se comporte el límite VC con m?

(b) Supongamos que p" E H", con una probabilidad a *priori* p . (En general, las p no son conocidas.)Dado queH He F1, '' *p* r21 . De qué componentes del problema de aprendizaje dependen los p 's.

(c) Supongamos que g" g E He . Demuéstrese que

$$\mathbb{P}\left[|E_{\text{in}}(g_i) - E_{\text{out}}(g_i)| > \epsilon \mid g^* = g_m\right] \leq \frac{1}{\cdot} \cdot 4m_{\mathcal{H}_i}(2N)e^{-\epsilon^2 N/8}$$

Aquí, el condicionamiento consiste en seleccionar la función .*[Sugerencia: Limite la probabilidad 0'* tmaxp¿g, |A, (g) - Aout(g) */c\g"-g"* *Utiliza el teorema de Boyes para reescr*$\frac{1}{p_i}\mathbb{P}[\max_{g\in\mathcal{H}_i}|E_{\text{in}}(g) - E_{\text{out}}(g)| >$ E y g" g]. *Utilice el hecho de que SQA* y *B)* *\y argumente que puede aplicar la conocida desigualdad VC a la expresión resultante.)*

Puede interpretar este resultado de la siguiente manera: si utiliza SRM y termina con g, , entonces el límite de generalización es un factor 1 peor que el límite que habría obtenido si hubiera empezado simplemente con H ; ése es el precio que paga por permitirse la posibilidad de buscar más que H",. Normalmente, los modelos más sencillos aparecen antes en la estructura, por lo que el límite será razonable si la función objetivo es sencilla (en cuyo caso p es grande (o m pequeño)). SRM funciona bien en la práctica.

Problema 8.16 Que puede plantearse dentro del marco SRM: selección entre diferentes restricciones de orden suave **(/Yg}gto** o selección entre diferentes parámetros de regularización **(Ht}ty** donde el conjunto de hipótesis se fija enH y el algoritmo de aprendizaje es de minimización de error aumentado con diferentes parámetros de regularización A.

Problema 8.17 Supongamos que utilizamos "SRM" para seleccionar entre un conjunto arbitrario de modelos H1, ... , H// con d"(J-tm-F1) *dv]I)* (frente a una estructura en la que se cumple la condición adicionalH C H +j).

(a) ¿Es posible (o *E"]I*) < E , (J-tm-F1)

(b) Sea p la probabilidad de que el proceso conduzca a una función g EH , con *p"* -- 1. Dé un límite para el error de generalización en términos de *d"(/Y*).

Problema 8.18 Supongamos que podemos ordenar las hipótesis de un modelo, H 1, 2, ...}. Supongamos que *d" I)* es infinito. Definir los subconjuntos de hipótesisH (L1, 2 , ... , in). Suponga que implementa un algoritmo de aprendizaje paraH como sigue: comience con in ; si C; (ñ) v , deténgase y obtenga in ; si no, pruebe hz: y así sucesivamente

(a) Supongamos que sale lip, por lo que efectivamente sólo ha buscado las m hipótesis en H",. ¿Puede utilizar el límite VC: (con alta probabilidad)

$E_{\text{out}}(h_m) \leq \nu + \sqrt{\frac{8\ln(2rn/6)y}{2N}}$ En caso afirmativo, ¿por qué? En caso negativo, ¿por qué no?

(b) Formule este proceso en el marco de la SRM. *|Pista: las* H", *'s forman una estructura).*

(c) ¿Puede llegar a alguna conclusión de generalización (recuerde, *d c(J-I)*)? I(sí, ¿cuál es el límite del error de generalización y cuándo espera una buena generalización? En caso negativo, ¿por qué?