3. Discriminantes lineales

Facundo Bromberg Ph.D.
Laboratorio DHARMa Dept. de Sistemas de Información
U. Tecnológica Nacional - Facultad Regional Mendoza - Argentina
fbromberg@frm.utn.edu.ar
http://dharma.frm.utn.edu.ar

Agosto 2012

Los contenidos del presente capítulo han sido desarrollados tomando como referencia principal la sección 4.1 de *Pattern Recognition and Machine Learning* de Christopher M. Bishop [1].

Índice

1. Funciones discriminantes lineales **2**
1.1. Discriminantes lineales binarios 2

2. Multi clases **5**

3. Aprendizaje de discriminadores lineales vía mínimos cuadrados **8**

4. Perceptrón **9**
4.1. Aprendizaje del Perceptron 11

5. Conclusiones finales **13**

6. Ejercicios **13**

1. Funciones discriminantes lineales

Presentamos aquí métodos para el aprendizaje de clasificadores cuyo espacio de *hypotesis* $\mathcal{H}$ es el conjunto de discriminantes lineales representados geometricamente como hyperplanos. Estos son un claro ejemplo del enfoque discriminitivo de aprendizaje que directamente aprende la función que discrimina los ejemplos de entrenamiento en cada una de sus clases, en contraste con el enfoque generativo, que primero aprende el modelo de generación de los datos (e.g., una distribución de probabilidad), y clasifica utilizando este modelo.

1.1. Discriminantes lineales binarios

Consideraremos primero el caso de clasificación binaria, donde el conjunto de datos de entrenamiento $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$ consiste en ejemplos de entrenamiento que pueden ser clasificados en solamente dos clases *positiva* y *negativa*, denotadas respectivamente $\mathcal{C}^+$ y $\mathcal{C}^-$. Nuestro espacio de hypotesis consistirá entonces en hyperplanos $D-1$-dimensionales, siendo D la dimensión del espacio de $\mathbf{x}$, que divide al espacio en dos semi-planos, correspondientes a las clases $\mathcal{C}^+$ y $\mathcal{C}^-$, respectivamente. La Figura 1 muestra un ejemplo para el caso de 2D, donde el separador lineal es una linea recta (i.e., un hyperplano de 1D).

El caso de discriminadores lineales es en sí un clasificador muy sencillo que en la práctica es poco usado. Sin embargo, el formalismo que involucra sirve de sustento para tanto para *Perceptrón*, *redes neuronales artificiales*, como así también para *Máquinas de Vectores Soporte* (SVM por sus siglas en inglés).

Formalmente, un hyperplano H puede definirse a través de su vector perpendicular $\mathbf{w}$ a través de la siguiente ecuación:

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{1}$$

que deben satisfacer todos los puntos $\mathbf{x}$ del espacio que pertenecen a H. Por razones que quedarán claras mas adelante, al vector $\mathbf{w}$ se le llama vector de *pesos*, y a b se le llama el *bias*.

Es facil ver que $\mathbf{w}$ debe ser perpendicular al hyperplano H. Tal como lo ilustra la Figura 2. la resta de dos puntos $\mathbf{x}_A$ y $\mathbf{x}_B$ pertenecientes a H es un vector que yace sobre el hyperplano. Dado que ambos pertenecen al hyperplano, tenemos que $\mathbf{w}^T\mathbf{x}_A + b = 0$ y $\mathbf{w}^T\mathbf{x}_B + b = 0$, y restando ambas ecuaciones tenemos $\mathbf{w}^T(\mathbf{x}_A - \mathbf{x}_B) = 0$, es decir, $\mathbf{w}$ es perpendicular a todo vector que yace en el hyperplano, y por lo tanto es perpendicular.

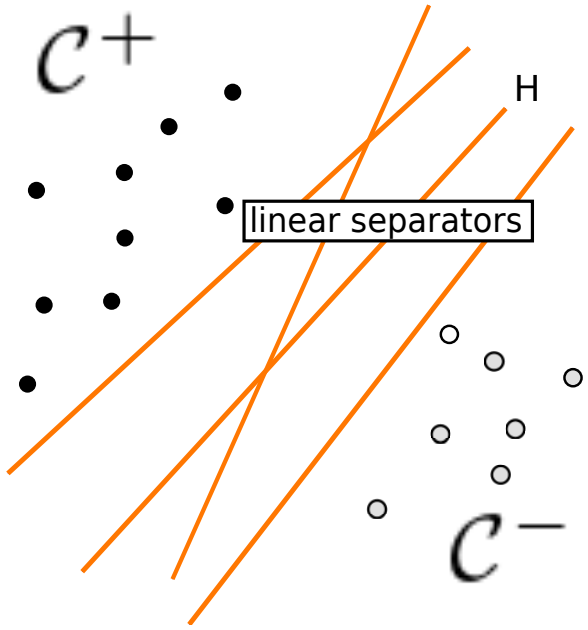

Figura 1: Separadores lineales de 1D (en naranja), en un espacio de entrada de 2D.

Del parrafo anterior vemos que $\mathbf{w}$, además de ser perpendicular al hyperplano, define una dirección. Llamamos $\mathcal{R}^+$ al semiplano al que apunta $\mathbf{w}$, y $\mathcal{R}^-$ al otro semiplano. De esta manera, el hyperplano H puede usarse como clasificador notando que

Lema 1.

$$\mathbf{x} \in \mathcal{R}^+ \iff y(\mathbf{x}) \geq 0$$
$$\mathbf{x} \in \mathcal{R}^- \iff y(\mathbf{x}) < 0$$

donde $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$. Así, todo punto $\mathbf{x} \in \mathcal{R}^+$ se clasifica en la clase $\mathcal{C}^+$, y todo punto $\mathbf{x} \in \mathcal{R}^-$ en la clase $\mathcal{C}^-$:

$$\begin{cases} \mathbf{x} \text{ se clasifica como perteneciente a } \mathcal{C}^+ & \text{si } y(\mathbf{x}) \geq 0 \\ \mathbf{x} \text{ se clasifica como perteneciente a } \mathcal{C}^- & \text{de lo contrario.} \end{cases}$$

Tal como lo ilustra la Figura 1, nuestro objetivo entonces consistirá, dado un conjunto de entrenamiento con los puntos pre-clasificados, encontrar el hyperplano H que resulte en todos los puntos pre-clasificados en la clase $\mathcal{C}^+$ ubicados en $\mathcal{R}^+$, y todos los puntos pre-clasificados en $\mathcal{C}^-$ ubicados en $\mathcal{R}^-$.

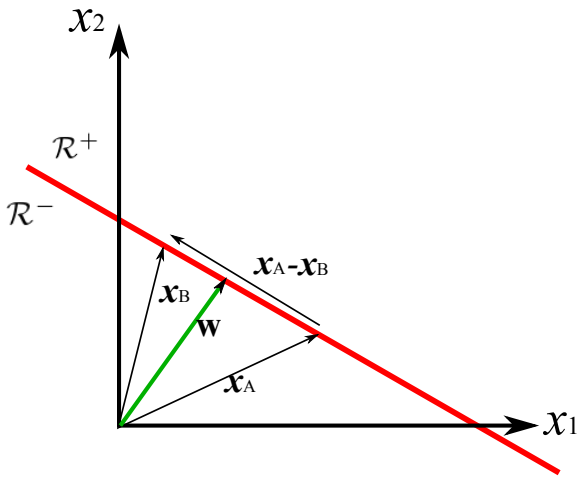

Figura 2: Geometría del hyperplano.

Para demostrar el Lema 1, determinaremos la distancia perpendicular $r(\mathbf{x})$ del punto $\mathbf{x}$ al hyperplano. Tal como lo muestra la Figura 3, el vector $\mathbf{x}$ puede descomponerse en la suma de dos vectores:

$$\mathbf{x} = \mathbf{x}_\perp + \frac{\mathbf{w}}{\|\mathbf{w}\|} r(\mathbf{x}) \tag{2}$$

donde el segundo vector apunta en la dirección perpendicular al hyperplano. Puede concluirse entonces que

$$\begin{aligned} \mathbf{x} \in \mathcal{R}^+ &\iff r(\mathbf{x}) \geq 0 \\ \mathbf{x} \in \mathcal{R}^- &\iff r(\mathbf{x}) < 0. \end{aligned} \tag{3}$$

De la descomposición de la Eq. (2) tenemos entonces que

$$\begin{aligned} y(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} + b \\ &= \mathbf{w}^T \left(\mathbf{x}_\perp + \frac{\mathbf{w}}{\|\mathbf{w}\|} r(\mathbf{x}) \right) + b \\ &= y(\mathbf{x}_\perp) + \frac{\mathbf{w}^T\mathbf{w}}{\|\mathbf{w}\|} r(\mathbf{x}) \\ &= r(\mathbf{x})\|\mathbf{w}\| \end{aligned}$$

El Lema 1 queda demostrado de la Eq.(3), notando que el signo de $y(\mathbf{x})$ corresponde al signo de $r(\mathbf{x})$ dado que $\|\mathbf{w}\|$ es siempre positivo.

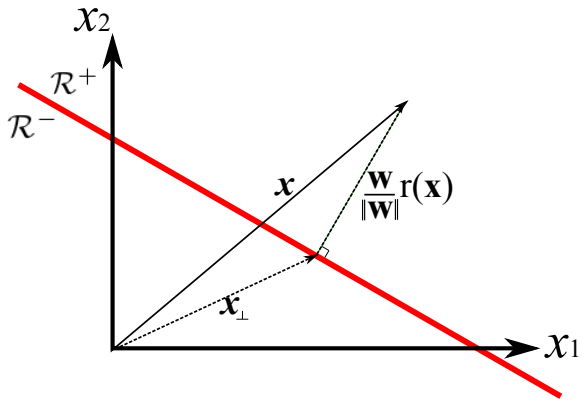

Figura 3: Geometría de la distancia perpendicular $r(\mathbf{x})$ del punto $\mathbf{x}$ al hyperplano.

2. Multi clases

Consideramos ahora la extensión de discriminantes lineales binarios al caso de $K > 2$ clases $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$. Uno estaría tentado a construir un discriminante de K-clases, simplemente combinando un cierto número de discriminantes binarios. Esto sin embargo lleva a importantes dificultades, como veremos a continuación.

Consideramos primero el caso denominado *one-versus-all*, ilustrado en la Fig. 4 (izquierda) para $K = 3$, donde se consideran $K - 1$ discriminantes binarios, donde el k-esimo discriminante separa los puntos entre los que pertenecen a $\mathcal{C}_k$ y los que no (es decir, pertenencen a $\neg\mathcal{C}_k$). Como muestra la figura, la región marcada en verde (y el signo ?) es ambigua ya que indica que sus puntos pertenecen tanto a $\mathcal{C}_1$ como a $\mathcal{C}_2$.

El segundo caso, ilustrado en Fig. 4 (derecha) y denominado *one-versus-one*, considera $\binom{K}{2} = K(K-1)/2$ discriminadores binarios, uno por cada posible par de clases, ejemplificado para $K = 3$. Cada punto es clasificado por voto mayoría entre los distintos discriminantes. Así, por ejemplo, en la región $\mathcal{R}_2$, hay dos votos por $\mathcal{C}_2$ y un voto por $\mathcal{C}_1$, con lo que los puntos se clasifican en $\mathcal{C}_2$, y de allí el nombre de la región. De la misma manera, en todas las demás regiones existe alguna clase con mayoría de votos, a excepción de la región marcada en verde (y el signo ?), donde cada una de las clases obtiene exactamente un voto, dejando ambigua la decisión sobre la clase.

Estas dificultades se resuelven generalizando los discriminadores lineales

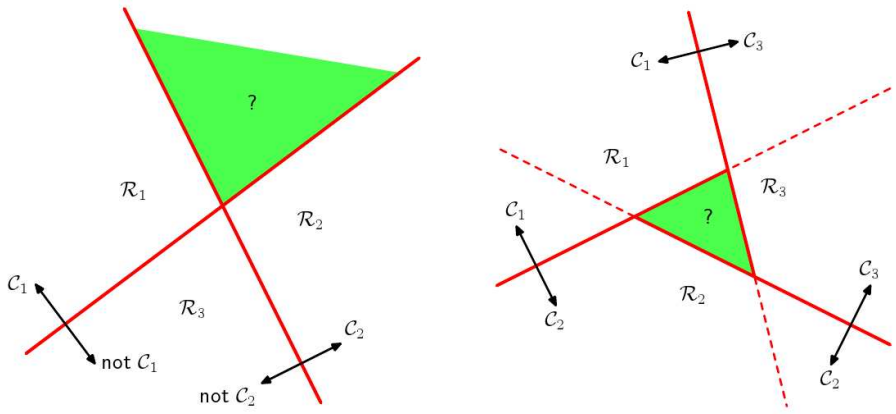

Figura 4: Intentos de construir un discriminante de K clases $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ a partir de un conjunto de de discriminantes binarios, ejemplificado para el caso de $K = 3$. Como muestra la figura, esto resulta en regiones ambiguas (en verde). En la izquierda se muestra el caso de $K - 1$ discriminantes binarios, cada uno discriminando entre $\mathcal{C}_k$ y $\neg\mathcal{C}_k$. $k = 1, \ldots, K - 1$. En la derecha se muestra el caso de $\binom{K}{2}$ discriminantes, discriminando entre clases $\mathcal{C}_j$ y $\mathcal{C}_k$, para $j \neq k$.

a discriminadores lineales por partes (piece-wise linear), ver Fig. 5, que consideran un solo discriminante multi-clase compuesto por K funciones lineales de la forma

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + b_k$$

el cual clasifica un punto $\mathbf{x}$ como perteneciente a la clase $\mathcal{C}_k$ si

$$y_k(\mathbf{x}) > y_j(\mathbf{x}) \text{ para todo } j \neq k; \tag{4}$$

por lo que comunmente se le denomina *one-versus-all*.

Veamos que este clasificador no resulta en regiones ambiguas. Para ello, asumimos, por vía del absurdo, que existe un punto $\mathbf{x}$ tal que $\mathbf{x} \in \mathcal{C}_j$ y $\mathbf{x} \in \mathcal{C}_{j'}$, para dos clases arbitrarias $\mathcal{C}_j$ y $\mathcal{C}_{j'}$. Entonces, por Eq. (4) y $\mathbf{x} \in \mathcal{C}_j$,

$$y_j(\mathbf{x}) > y_k(\mathbf{x}), \text{ para todo } j \neq k;$$

y en particular, para $k = j'$, tenemos que $y_j(\mathbf{x}) > y_{j'}(\mathbf{x})$. Además, por por Eq. (4) y $\mathbf{x} \in \mathcal{C}_{j'}$

$$y_{j'}(\mathbf{x}) > y_k(\mathbf{x}), \text{ para todo } j' \neq k;$$

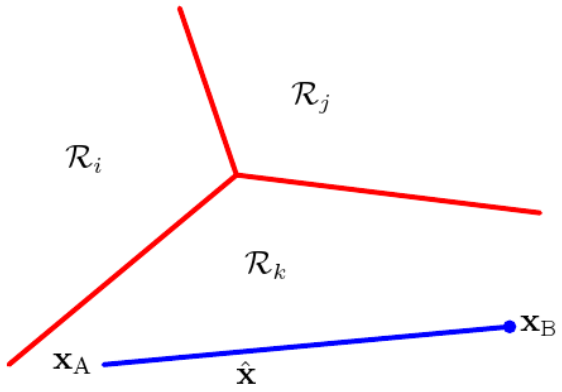

Figura 5: Regiones de decisión para un discriminante multi-clase ($K = 3$). La figura ilustra el hecho de que estas regiones son convexas y simplemente conectadas, es decir, que todo punto $\widehat{\mathbf{x}}$ que yace en la recta que une dos puntos $\mathbf{x}_A$ y $\mathbf{x}_B$ pertenecientes a una misma región, debe pertenecer a esa misma región.

en particular, para $k = j$, tenemos que $y_{j'}(\mathbf{x}) > y_j(\mathbf{x})$, un absurdo.

Para concluir, veamos un poco las características geométricas de este clasificador. La superficie de separación entre las clases $\mathcal{C}_k$ y $\mathcal{C}_j$ esta claramente dada por todos los puntos $\mathbf{x}$ que satisfacen $y_k(\mathbf{x}) = y_j(\mathbf{x})$, y por lo tanto corresponde a un hyperplano $(D-1)$-dimensional definido por

$$(\mathbf{w}_k - \mathbf{w}_j)^T\mathbf{x} + (b_k - b_j) = 0,$$

con todas las propiedad geometricas ya descriptas para el caso binario.

Además, es posible demostrar que las regiones de decisión son convexas y simplemente conectadas, tal como lo ilustra la Fig. 5 para $K = 3$. Para ello, es suficiente demostrar que todo punto $\widehat{\mathbf{x}}$ que yace en la linea que conecta dos puntos $\mathbf{x}_A$ y $\mathbf{x}_B$, ambos pertenecientes a la misma región $\mathcal{R}_k$, también pertenece a $\mathcal{R}_k$. La recta que une $\mathbf{x}_A$ y $\mathbf{x}_B$ está definida por

$$\widehat{\mathbf{x}} = \lambda\mathbf{x}_A + (1-\lambda)\mathbf{x}_B$$

con $0 \leq \lambda \leq 1$. Por la linearidad de las funciones discriminantes tenemos que

$$y_k(\widehat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1-\lambda)y_k(\mathbf{x}_B).$$

Como ambos $\mathbf{x}_A$ y $\mathbf{x}_B$ yacen en $\mathcal{R}_k$, sigue que $y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A)$, y $y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$, para todo $j \neq k$, y por lo tanto $y_k(\widehat{\mathbf{x}}) > y_j(\widehat{\mathbf{x}})$, es decir, $\widehat{\mathbf{x}}$ yace en $\mathcal{R}_k$.

3. Aprendizaje de discriminadores lineales vía mínimos cuadrados

Tal como lo ilustra la Figura 1 para el caso binario de $K = 2$, el aprendizaje de un discriminador lineal, dado un conjunto de entrenamiento $\mathcal{T}$ con los puntos pre-clasificados, es encontrar el hyperplano H que resulte en todos los puntos pre-clasificados en la clase $\mathcal{C}_j$ ubicados en $\mathcal{R}_j$. A este tipo de hypotesis la llamamos *hypotesis consistente*.

Para ello, tomaremos el enfoque general de minimizar una función de perdida o error $E_{\mathcal{T}}(H)$ (*loss function*) dependiente de la hypotesis H, de tal manera que esta función se minimize cuando *todos* los ejemplos de entrenamiento en $\mathcal{T}$ han sido correctamente clasificados por H, i.e., el error se minimize cuando H es consistente. En nuestro caso, consideraremos como función error el *error cuadratico medio*.

Interesantemente, para el caso de discriminadores lineales, la minimización puede realizarse en forma analítica, resultando en una expresión que determina H en función de $\mathcal{T}$. Para ello, introducimos un poco de notación.

Para comenzar, consideraremos una codificación llamada *1-de-K* para la variable de clase t, de tal manera que t se convierte en un vector, denotado por $\mathbf{t}$, con cardinalidad K, de tal manera que si t corresponde a la k-esima clase $\mathcal{C}_k$, entonces el vector $\mathbf{t}$ tendra todas sus componentes iguales a 0, a excepción de la k-esima que tendrá el valor 1. Por ejemplo, si t corresponde a $\mathcal{C}_3$, y $K = 6$, entonces $\mathbf{t} = (0, 0, 1, 0, 0, 0)$.

Con esta notación en mano podemos expresar el error-cuadrático-medio de la siguiente manera

$$E_{\mathcal{T}}(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \frac{1}{2} \sum_{k=1}^{K} \sum_{n=1}^{N} (y_k(\mathbf{x}_n) - t_k)^2. \tag{5}$$

Además, consideraremos que cada vector de pesos $\mathbf{w}_k$ es extendido en un vector $\widetilde{\mathbf{w}} = (\mathbf{w}_k, b_k)$ con una nueva componente que siempre vale b_k, y cada vector de entrada $\mathbf{x}$ es extendido en un nuevo vector $\widetilde{\mathbf{x}} = (\mathbf{x}, 1)$ con una nueva componente que siempre vale 1. De esta manera, resulta para todo k que

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + b_k = \widetilde{\mathbf{w}}^T \widetilde{\mathbf{x}}$$

Luego, reformulamos a $\mathcal{T}$ y H en términos matriciales. Comenzamos agrupando las funciones discriminates en forma matricial

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^T \widetilde{\mathbf{x}},$$

donde $\widetilde{\mathbf{W}}$ es una matriz cuya k-esima columna corresponde a $\widetilde{\mathbf{w}}_k$, y donde las K componentes de $\mathbf{y}$ corresponden a las funciones lineales y_k, $k = 1, \ldots, K$. Así mismo, consideraremos la matrix $\mathbf{T}$, cuyo n-esimo renglón corresponde al n-esimo vector de clase $\mathbf{t}_n$; y la matrix $\widetilde{\mathbf{X}}$, cuyo n-esimo renglón es $\widetilde{\mathbf{x}}_n^T$.

Con esta notación, la función del error cuadrático medio puede escribirse de la siguiente manera:

$$E_{\mathcal{T}}(\widetilde{\mathbf{W}}) = \frac{1}{2} Tr\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^T(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}. \tag{6}$$

Finalmente, fijando las derivadas respecto a $\widetilde{\mathbf{W}}$ igual a cero y reacomodando, obtenemos la solución para $\widetilde{\mathbf{W}}$ (que define el hyperplano solución H) como

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{T}.$$

4. Perceptrón

El Perceptrón es otro ejemplo de discriminante lineal (binarios) propuesto por Rosenblat en 1957 [2] y ocupa un importante lugar en la historia de los algoritmos de reconocimiento de patrones. Su principal contribución es la posibilidad de transformar no-linealmente el vector de entrada $\mathbf{x}$ (de D dimensiones) a un vector $\phi(\mathbf{x})$ de M dimensiones (comunmente $M > D$). Con esta transformación, un conjunto de entrenamiento $\mathcal{T} = \{(\mathbf{x}_1, t_1), \cdots, (\mathbf{x}_N, t_N)\}$ que *no* es linealmente separable puede convertirse en un conjunto linealmente separable $\{(\phi(\mathbf{x})_1, t_1), \cdots, (\phi_N(\mathbf{x}), t_N)\}$ (con las mismas etiquetas de clase) en el espacio proyectado. Veamos un ejemplo

Example 1. *El ejemplo considera la transformación* $\mathbf{x} = (x_1, x_2) \longrightarrow \phi = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. *Como antes, la superficie de separación debe satisfacer* $\mathbf{w}^t.\phi(\mathbf{x}) + b = 0$, *es decir,* $w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 + b = 0$. *Esta última ecuación es claramente un hiperplano en 3D (en el espacio proyectado* ϕ*), pero en el espacio* x *corresponde a una ecuación de una sección cónica* $Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F = 0$, *con* $D = E = 0$, $A = w_1$, $B = w_2\sqrt{2}$, $C = w_3$, *y* $F = b$, *correspondiente a un circulo, elipse, parábola o hiperbola dependiendo de los valores de* A, B, C *y* F, *i.e., en los valores de* $\mathbf{w}$ *y* b. *Esto es ilustrado graficamente en la Fig.(6), donde un dataset que no es separable linealmente en el espacio* $\mathbf{x}$, *puede ser separado linealmente en el espacio* ϕ *(derecha), correspondiente a un circulo en el espacio* $\mathbf{x}$ *(izquierda).*

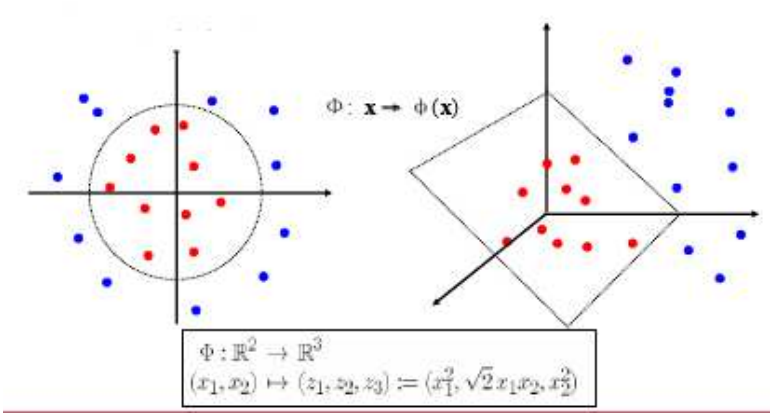

Figura 6: Una muestra que no es linealmente separable en el espacio $\mathbf{x}$ (izquierda), es separada linealmente en el espacio ϕ (derecha); donde el hiperplano en ϕ (derecha) se corresponde a una superficie de separación no-lineal (un circulo) en el espacio $\mathbf{x}$ (izquierda)

.

Además de proponer la transformación no-lineal ϕ, el perceptrón propone una función discriminante no-lineal

$$y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x})), \tag{7}$$

a través de la *función de activación* no-lineal f, dada por una función escalon de la forma

$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1 & a < 0. \end{cases}$$

y donde el *bias* b aparece implicito en el vector de pesos $\mathbf{w}$, y el vector $\phi(\mathbf{x})$ incluye una componente igual a 1.

Este esquema matemático puede representarse graficamente como lo ilustra la Fig. 7. Esta forma de representarlo proviene de la inspiración inicial del Perceptrón, la cual tuvo raices neuro-biológicas. Aquí vemos que las entradas $x_1, \ldots, x_D$ entran a través de las neuronas de entrada. El i-esimo input x_i se propaga hacia la neurona central (en rosado) donde en las sinapsis es re-escaleado linealmente por la eficiencia sináptica, modelada como un producto con el peso sináptico w_i. Los inputs pesados (incluyendo el bias b) convergen en el núcleo de la neurona donde la modelización mas simple consiste en sumarlos. Luego, de superar esta sumatoria un umbral, la neurona dispara (modelado como salida $+1$), o de lo contrario se inactiva (salida -1).

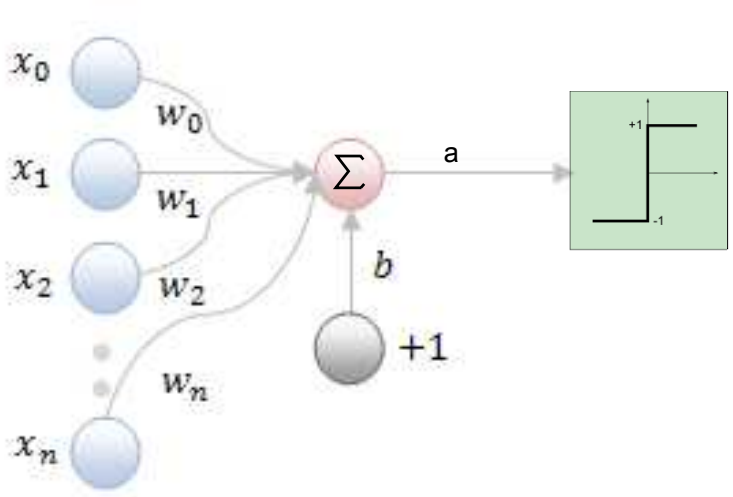

Figura 7: Representación gráfica del perceptrón.

4.1. Aprendizaje del Perceptron

Al igual que en el caso de separadores lineales, propondremos una función de pérdida o error y propondremos como solución al conjunto de pesos $\mathbf{w}$ que la minimice. Sin embargo, debido a que la función de activación es no-lineal, no es posible proponer aquí el mismo esquema de resolución analítica considerado para separadores lineales.

Otra de las importantes contribuciones de Rosenblat fué proponer una función de perdida que garantice la convergencia de algoritmos de optimización numérica (consideraremos aquí el algoritmo de gradiente descendiente, ver mas adelante). La función de perdida propuesta por Rosenblat se le llama *criterio del perceptrón*. Para derivarla, notamos primero que buscamos un vector de pesos $\mathbf{w}$ tal que inputs $\mathbf{x}_n$ en $\mathcal{C}^+$ tendrán $\mathbf{w}^T\phi(\mathbf{x}_n) > 0$ e inputs $\mathbf{x}_n$ en $\mathcal{C}^-$ tendrán $\mathbf{w}^T\phi(\mathbf{x}_n) < 0$. Usando como codificación de las etiquetas de clase $t_n\{+1, -1\}$ (a diferencia de la codificación 1-en-K considerada para separadores lineales), vemos que buscamos entonces que cualquier input, independientemente de la clase a la que pertenezca, satisfagan $\mathbf{w}^T\phi(\mathbf{x}_n)t_n > 0$. En cambio, si para cierto $\mathbf{w}$ un input $\mathbf{x}_n$ es clasificado erroneamente, el signo de t_n no corresponderá con el signo de su activación $\mathbf{w}^T\phi(\mathbf{x}_n)$, resultando en $\mathbf{w}^T\phi(\mathbf{x}_n)t_n < 0$.

El criterio del perceptrón entonces penaliza *solamente* a los inputs $\mathbf{x}_n$

que han sido erroneamente clasificados:

$$E_P(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^T \phi_n t_n$$

donde $\phi_n = \phi(\mathbf{x}_n)$ y $\mathcal{M}$ denota el conjunto de inputs clasificados erroneamente.

Para encontrar $\mathbf{w}$ solución, minimizamos esta función error usando el algoritmo de gradiente descendiente, que indica que los pesos deben modificarse en dirección contraria al gradiente (el cual apunta en la dirección de máxima pendiente):

$$\begin{aligned} \mathbf{w}^{\tau+1} &\longleftarrow \mathbf{w}^{\tau} - \eta \nabla E_P(\mathbf{w}^{\tau}) \\ &= \mathbf{w}^{\tau} - \eta \sum_{n\in\mathcal{M}} \phi_n t_n. \end{aligned} \tag{8}$$

donde η es la *taza de aprendizaje*, y η es un entero que indexa los pasos de algoritmo. Este régimen de actualización de pesos se llama en la práctica *aprendizaje por lotes* ya que actualiza los pesos en base a *todos* los errores en el dataset de entrenamiento. Una alternativa, es actualizar los pesos con el error de cada ejemplo a la ves:

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \eta \phi_n t_n.$$

lo que en la práctica resulta en mejores convergencias.

Una interpretación sencilla del algoritmo es la siguiente. Se cicla sobre los inputs de entrenamiento, y para cada input $\mathbf{x}_n$ evaluamos la función del perceptrón $y(\mathbf{x}_n)$ de Eq. (7). Si el patrón es clasificado correctamente (i.e., $y(\mathbf{x}_n) = t_n$), los pesos no cambian. En cambio, si es clasificado incorrectamente, entonces, si su clase es $\mathcal{C}^+$ sumamos ϕ_n al vector de pesos, sino restamos ϕ_n. Esto es ilustrado en la Fig. 8.

Claramente, es importante asegurarse que el algoritmo convergerá eventualmente a una solución que separe el conjunto de entrenamiento. Existe el *teorema de convergencia del perceptrón*, que garantiza que se econtrará la solución siempre que esta exista, es decir, siempre que el conjunto de entrenamiento mapeado al espacio ϕ sea linealmente separable. Es importante notar que, de no ser separable el dataset, el algoritmo puede no converger nunca.

Incluso si los datos son linealmente separable, pueden haber muchas soluciones, y a cual converja el algoritmo depende de los parámetros iniciales y el orden en que se presentan los datapoints.

5. Conclusiones finales

Como adelantamos al comienzo de este capítulo, el estudio de los discriminadores lineales tiene como propósito motivar el estudio de formalizmos mas útiles en la práctica como son las redes neuronales o las máquinas de vectores soporte. Pero, ¿porqué es que los discriminadores lineales no sirven en la práctica?. La respuesta mas inmediata y obvia es que en muchos casos prácticos, los datasets no son linealmente separables. Como vimos, el Perceptrón introduce un truco para resolver este problema que consiste simplemente en el mapeo del dataset a un espacio de mayores dimensiones. Sin embargo, este mapeo conlleva dos dificultades importantes. Una de ellas es que es extremadamente costoso demostrar que no son linealmente separables, con lo que es imposible garantizar que el mapeo propuesto resulte en un dataset linealmente separable. Esta dificultad es resuelta tanto por las redes neuronales artificiales como por las máquinas de vectores soporte. En el caso de las redes neuronales, estas parametrizan el mapeo y encuentran la parametrización dinamicamente durante el entrenamiento, garantizando así la separabilidad del mapeo resultante. Sin embargo, las redes neuronales son afectadas por la segunda dificultad que conlleva el mapeo a muchas dimensiones: el sobre-ajuste. Este problema surje de la infinidad de soluciones consistentes (i.e., que separan correctamente al dataset) muchas de las cuales, si bien separan correctamente al dataset de entrenamiento, pueden resultar en clasificaciones muy pobres para datos novedosos. Mas aún, este problema es exacerbado para altas dimensiones (ver detalles en capítulos posteriores). Mínimos cuadrados ni el Perceptrón hacen un buen trabajo al respecto, y si bien existen otras funciones de perdida como el discriminante lineal de Fisher (ver [1], sección 4.1.4) que ayudan en este respecto, es recien con el advenimiento de las máquinas de vectores soporte que se encontró un método eficiente y con garantías teóricas para encontrar la hypotesis consistente que minimize el error de generalización.

6. Ejercicios

1. Dado un conjunto de datos $\{\mathbf{x}_n\}$, definimos su casco convexo (*convex hull*) como el conjunto de todos los puntos determinados por

$$\mathbf{x} = \sum_n \alpha_n \mathbf{x}_n$$

y $\alpha_n \geq 0$ y $\sum_n \alpha_n = 1$. Considere también otro conjunto de puntos $\{\mathbf{y}_n\}$ junto con su correspondiente casco convexo. Por definición, estos

dos conjuntos de puntos serán linearlmente separables si existe un vector $\widehat{\mathbf{w}}$ y un escalar b tal que $\widehat{\mathbf{w}}^T\mathbf{x}_n + b > 0$ y $\widehat{\mathbf{w}}^T\mathbf{y}_n + b < 0$. Demuestre que si sus cascos convexos intersectan (i.e., tienen puntos en común), los dos conjuntos de puntos no pueden ser linearlmente separables, y, a la inversa, si son linealmente separables, sus cascos convexos no pueden intersectar.

2. Demuestre la equivalencia entre las Eqs. (5) y (6).

3. Demuestre a pesar de estar modulado por una función de activación no-lineal f, la función de separación $y(\mathbf{x}) = f(\mathbf{w}^T\phi(\mathbf{x}) + b)$ también resulta en un separador lineal en el espacio de features ϕ. (Ayuda: considerar la condición que debe satisfacer la activación $\mathbf{w}^T\phi(\mathbf{x}) + b$ para los puntos que yacen sobre el separador.$y(\mathbf{x}) > 0$)

Referencias

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

[2] F. Rosenblatt. The perceptron–a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

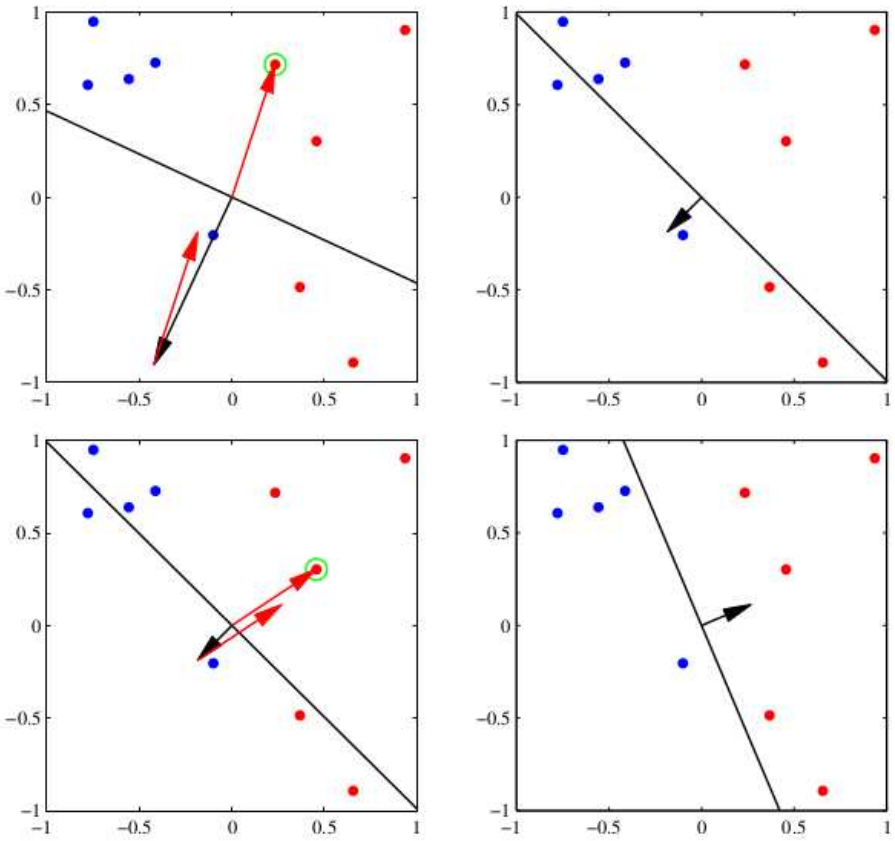

Figura 8: Traza del algoritmo de aprendizaje del perceptrón que demuestra su convergencia para un dataset binario (clases en rojo y azul), en un espacio 2D. El vector de pesos es representado por una flecha negra, y la superficie de decisión por una recta negra (con la flecha apuntando hacia la clase roja). Inicialmente, el datapoint marcado con el circulo verde se encuentra mal clasificado, con lo que su vector ϕ_n es sumado al vector de pesos, resultando en la nueva superficie de decisión mostrada en la esquina superior derecha. El siguiente renglón (izquierda) muestra el siguiente error (también en circulo verde), y el resultado de sumarlo en la figura de la derecha. El resultado es una recta que separa correctamente todos los ejemplos.