

MÁQUINAS DE VECTORES DE SOPORTE (SVM)

ALAN REYES-FIGUEROA

INTRODUCCIÓN A LA CIENCIA DE DATOS

(AULA 33) 18.MAYO.2022

Modelo Perceptrón

Introducido por F. Rosenblatt (1958), en *Psychological Review* Vol. **65**(6).

(\mathbb{X}, \mathbf{y}) conjunto de datos $\mathbb{X} \in \mathbb{R}^{n \times d}$. Codificamos las categorías $y_i \in \{-1, 1\}$.

Buscamos clasificadores lineales de la forma

$$\hat{y}(\mathbf{x}) = \text{sign}(g(\mathbf{x})) = \text{sign}(w_0 + \mathbf{w}^T \mathbf{x}),$$

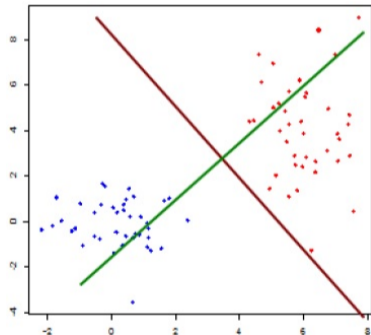
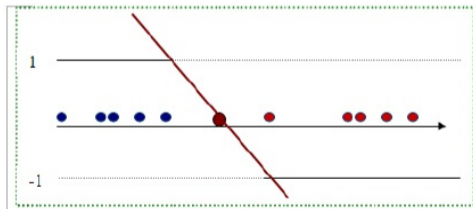
con $w_0 \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^d$.

Obs:

- La frontera de decisión es una curva de nivel de $g(\mathbf{x})$.
- \mathbf{w} indica la dirección (es el vector normal al hiperplano separante).
- w_0 indica la curva de nivel a elegir.

Modelo Perceptrón

Ejemplo: En \mathbb{R}^2 , $\mathbf{w} \in \mathbb{R}^2$ marca la línea verde (dirección normal al plano). La línea roja marca la frontera. Las curvas de nivel de $G(\mathbf{x})$ son paralelas a la línea roja.



Modelo Perceptrón

¿Cómo hallar w_0 y \mathbf{w} óptimos?

Vamos a definir una función de costo $C(w_0, \mathbf{w})$ (medida del desempeño), y vamos a minimizar dicha función.

Supongamos por ahora que los datos son linealmente separables.

- Primer intento: tomar el error empírico

$$C(w_0, \mathbf{w}) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[\text{sign}(g(\mathbf{x}_i)) \neq y_i].$$

Problema: C no es siempre derivable en w_0, \mathbf{w} ; y el gradiente muchas veces no es informativo.

¿Cómo construir funciones de costo más informativas?

Modelo Perceptrón

Tomemos ℓ el hiperplano de decisión, nos gustaría incluir en la función de costo un término que mida la distancia $d(\mathbf{x}, \ell)$ entre cada punto \mathbf{x} (mal clasificado) y ℓ .

Sea \mathbf{x}_ℓ la proyección ortogonal de \mathbf{x} a ℓ . Entonces

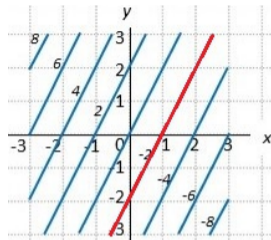
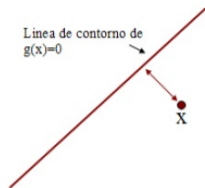
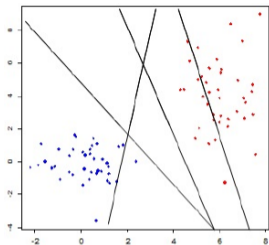
$$\mathbf{x} = \mathbf{x}_\ell + d \frac{\mathbf{w}}{\|\mathbf{w}\|}, \quad \text{con } d = d(\mathbf{x}, \ell) = |\mathbf{w}^T(\mathbf{x} - \mathbf{x}_\ell)|.$$

Multiplicando ambos lados por \mathbf{w}^T , y sumando w_0 , resulta

$$g(\mathbf{x}) = w_0 + \mathbf{w}^T \mathbf{x} = w_0 + \mathbf{w}^T \mathbf{x}_\ell + d \frac{\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} = g(\mathbf{x}_\ell) + d \|\mathbf{w}\| = d \|\mathbf{w}\|.$$

De ahí que $d = \frac{g(\mathbf{x})}{\|\mathbf{w}\|}$ y $d(\mathbf{x}, \ell) = \frac{|g(\mathbf{x})|}{\|\mathbf{w}\|}$.

Modelo Perceptrón



- Segundo intento: Por construcción, observe que \mathbf{x}_i está mal clasificado si $g(\mathbf{x}_i) y_i < 0$. Con base en esto, proponemos

$$C(\mathbf{w}_0, \mathbf{w}) = \sum_{i: g(\mathbf{x}_i) y_i < 0} \frac{|g(\mathbf{x}_i)|}{\|\mathbf{w}\|} = - \sum_{i: g(\mathbf{x}_i) y_i < 0} \frac{g(\mathbf{x}_i) y_i}{\|\mathbf{w}\|}.$$

Modelo Perceptrón

$$\text{Ahora, } C(w_0, \mathbf{w}) = - \sum_{i:g(\mathbf{x}_i) y_i < 0} \frac{g(\mathbf{x}_i) y_i}{\|\mathbf{w}\|} \text{ y } \|\mathbf{w}\| C(w_0, \mathbf{w}) = - \sum_{i:g(\mathbf{x}_i) y_i < 0} g(\mathbf{x}_i) y_i$$

tienen el mismo mínimo. Preferimos trabajar con

$$C_n(w_0, \mathbf{w}) = - \sum_{i:g(\mathbf{x}_i) y_i < 0} g(\mathbf{x}_i) y_i.$$

Haciendo el mapeo de \mathbb{R}^d a \mathbb{R}^{d+1} , dado por $\mathbf{x} \rightarrow (1, \mathbf{x})$, y escribiendo $\mathbf{w} = (w_0, \dots, w_d) \in \mathbb{R}^{d+1}$, tenemos

$$C_n(\mathbf{w}) = - \sum_{i:g(\mathbf{x}_i) y_i < 0} (\mathbf{x}_i^T \mathbf{w}) y_i, \quad \nabla_{\mathbf{w}} C_n(\mathbf{w}) = - \sum_{i:g(\mathbf{x}_i) y_i < 0} y_i \mathbf{x}_i.$$

Modelo Perceptrón

Tenemos el siguiente

Algoritmo: (Perceptrón, gradiente *online*)

- 1.) Inicio: Elegir $\alpha > 0$, $\mathbf{w}^{(0)} \in \mathbb{R}^{d+1}$ arbitrario.
- 2.) Repetir para $k = 0, 1, 2, \dots$ (hasta cierto criterio de paro):
 - Para cada dato mal clasificado \mathbf{x}_i hacer

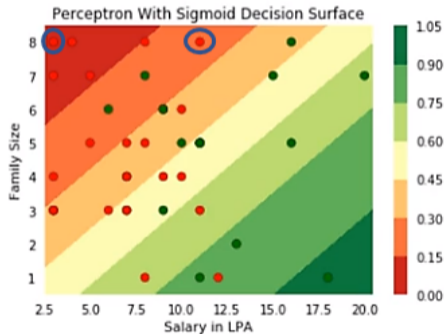
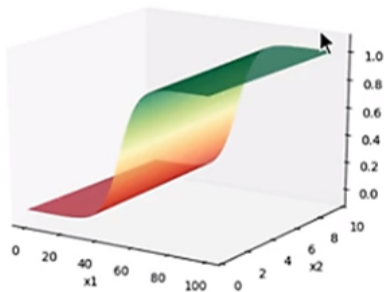
$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha \nabla_{\mathbf{w}} C_n(\mathbf{x}_i) = \mathbf{w}^{(k)} + \alpha y_i \mathbf{x}_i.$$

Obs!

- Si los datos son linealmente separables, habrá convergencia en tiempo finito.
- Cuidado con los mínimos locales de $C_n(\mathbf{w})$!
- <https://lecture-demo.ira.uka.de/neural-network-demo/?preset=Rosenblatt%20Perceptron>

Modelo Perceptrón

El modelo perceptrón y el modelo de clasificación logística son similares: ambos usan un gradiente en la dirección de \mathbf{w} , sin embargo la regresión logística pondera el gradiente mediante la función sigmoide σ .



Otros clasificadores lineales

Algunos modelos lineales resuelven el problema de clasificación como si fuera regresión:

- regresión logística.
- Se puede demostrar que LDA (análisis discriminante lineal) resuelve un problema de regresión (y también un *optimal scoring problem*) como:

$$\operatorname{argmin}_{\mathbf{w}} \sum_i (\vartheta(y_i) - \mathbf{w}^T \mathbf{x}_i)^2,$$

donde

$$\vartheta(y) = \begin{cases} -\frac{n}{n_-}, & \text{si } y = -1; \\ \frac{n}{n_+}, & \text{si } y = +1. \end{cases}$$

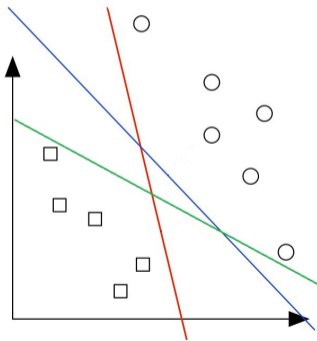
n_- y n_+ el número de observaciones negativas y positivas, resp.

Máquinas de Vectores de Soporte

Consideremos la siguiente situación: Aplicamos nuestro clasificador lineal

$$\hat{y}(\mathbf{x}) = \text{sign}(w_0 + \mathbf{w}^T \mathbf{x})$$

a un conjunto linealmente separable. Obtenemos varias soluciones (azul, rojo, verde).



Máquinas de Vectores de Soporte

Tenemos $C(\text{rojo}) = 0$, $C(\text{azul}) = 0$, $C(\text{verde}) = 0$.

Hemos evaluado igual cada una de las soluciones. ¿Cuál es mejor?

Máquinas de vectores de soporte:

Support vector machines (SVM). Introducidas por Vladimir Vapnik, 1970s (Papers en 1992, 1995a, 1995b).

Definición

El **margen** de un clasificador lineal es la menor distancia entre una observación \mathbf{x}_i y la frontera de decisión ℓ .

Idea: buscar clasificador lineal con margen máximo

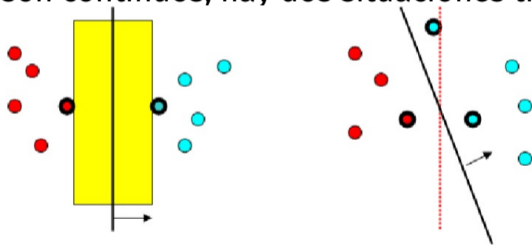
$$\operatorname{argmax}_{\mathbf{w}} C(\mathbf{w}), \quad \text{sujeto a } \frac{g(\mathbf{x}_i) y_i}{\|\mathbf{w}\|} \geq C, \quad \forall i.$$

Máquinas de Vectores de Soporte

Definición

Los **vectores de soporte** son las observaciones \mathbf{x}_i que están a una distancia mínima (igual al margen) de la frontera de decisión.

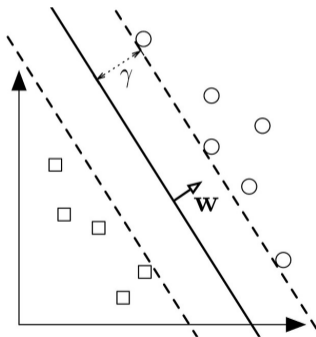
En 2D si los datos son continuos, hay dos situaciones típicas:



(a) 2 vectores de soporte, el margen es $\frac{1}{2}$ (distancia entre ellos); (b) 3 vectores de soporte. En \mathbb{R}^2 , si hay 4 vectores de soportes o más habrá colinealidad.

Máquinas de Vectores de Soporte

Si cambiamos $g(\cdot)$ por $cg(\cdot)$; $c > 0$, el clasificador $\hat{y}(\mathbf{x})$ no cambia. Podemos suponer que g en los puntos más cercanos a la frontera de decisión toma valor 1 ó -1 . En ese caso, el margen es $\gamma = C = \frac{1}{\|\mathbf{w}\|}$, $\mathbf{w} \in \mathbb{R}^d$.



Máquinas de Vectores de Soporte

Entonces, el problema de optimización a resolver

$$\operatorname{argmax}_{w_0, \mathbf{w}} C, \quad \text{sujeto a } \frac{g(\mathbf{x}_i) y_i}{\|\mathbf{w}\|} \geq C, \forall i.$$

se convierte en

$$\operatorname{argmax}_{w_0, \mathbf{w}} \frac{1}{\|\mathbf{w}\|}, \quad \text{sujeto a } g(\mathbf{x}_i) y_i \geq 1, \forall i,$$

o equivalentemente

$$\operatorname{argmin}_{w_0, \mathbf{w}} \|\mathbf{w}\|^2, \quad \text{sujeto a } g(\mathbf{x}_i) y_i \geq 1, \forall i,$$

Máquinas de Vectores de Soporte

Así, la formulación *primal* de las SVMs es

$$\operatorname{argmax}_{w_0, \mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w}, \quad \text{sujeto a } y_i(w_0 + \mathbf{w}^T \mathbf{x}_i) \geq 1, \quad \forall i. \quad (1)$$

Observe que este es un problema de programación cuadrática (problema de optimización con función objetivo cuadrática y restricciones lineales).

En la práctica, en lugar de resolver el problema primal, se resuelve el *problema dual* (ver Bishop, o Hastie, Tibshirani, Friedmann):

$$\operatorname{argmin}_{\alpha} \frac{1}{2} \alpha^T Q \alpha - \mathbf{1}^T \alpha, \quad \text{sujeto a } \mathbf{y}^T \alpha \geq 0, \quad \mathbf{y} \geq 0, \quad (2)$$

donde $Q = (Q_{ij})$, $q_{ij} = y_i \mathbf{x}_i^T \mathbf{x}_j y_j$.

Máquinas de Vectores de Soporte

Supongamos ahora que los datos no son linealmente separables.

Antes se exigía $\frac{g(\mathbf{x}_i) y_i}{\|\mathbf{w}\|} \geq C, \forall i$. En lugar de eso, se introducen ahora variables $\varepsilon_i \geq 0$, y se cambia lo anterior por

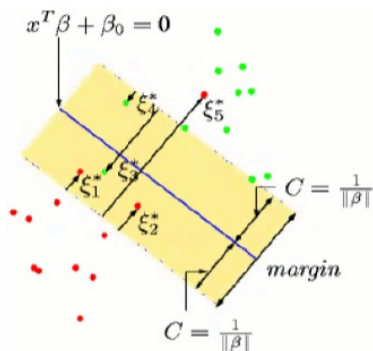
$$\frac{g(\mathbf{x}_i) y_i}{\|\mathbf{w}\|} \geq C(1 - \varepsilon_i), \quad \forall i.$$

Al mismo tiempo se castigan cada $\varepsilon_i \neq 0$, incluyéndolos en la función de costo, con un factor de penalización $\gamma > 0$. El problema a resolver resulta

$$\operatorname{argmax}_{\mathbf{w}_0, \mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_i \varepsilon_i, \quad \text{sujeito a } y_i(\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}_i) \geq 1 - \varepsilon_i, \quad \forall i. \quad (3)$$

Máquinas de Vectores de Soporte

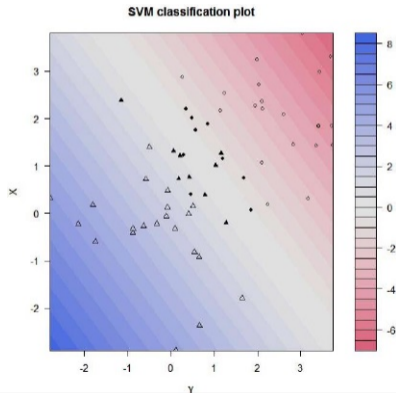
Interpretación de (3)



- ϵ_i es la distancia mínima que hay que trasladar \mathbf{x}_i para que esté al lado correcto de la frontera de decisión, y al menos a una distancia de $\frac{1}{\|\mathbf{w}\|}$.
- Si $g(\mathbf{x}_i) y_i < 0$ ó $0 < g(\mathbf{x}_i) y_i < 1$, tomamos ϵ_i tal que $g(\mathbf{x}_i) y_i = 1 - \epsilon_i$.
- Si $g(\mathbf{x}_i) y_i \geq 1$, tomamos $\epsilon_i = 0$.

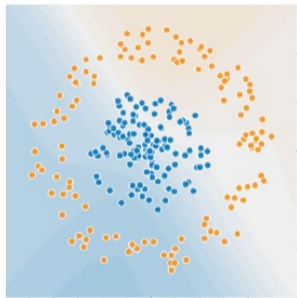
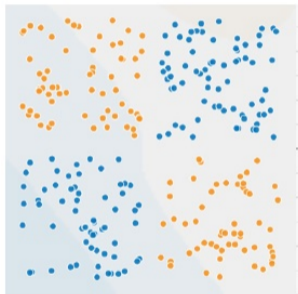
Máquinas de Vectores de Soporte

Se puede mostrar que la solución es de la forma $g(\mathbf{x}) = \sum_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + w_0$, donde sólo algunas α_i son diferentes de 0: aquellas que corresponden a observaciones con $\varepsilon_i \neq 0$ (los llamados vectores de soporte).



Caso general

Hasta ahora trabajamos con un clasificador lineal $\hat{y}(\mathbf{x}) = \text{sign}(w_0 + \mathbf{w}^T \mathbf{x})$.
¿Qué hacer si tenemos siguientes datos?



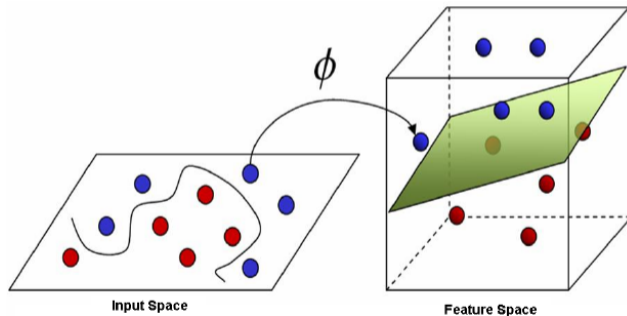
Ejemplos de datos no linealmente separables.

<http://playground.tensorflow.org/>

Caso general

Recordemos el **truco del kernel** (*kernel trick*): mapeamos los datos a un espacio de mayor dimensión, donde esperamos que sean linealmente separables.

Idea: transformar (implícitamente) los datos: $\mathbf{x} \rightarrow \Phi(\mathbf{x})$.



Caso general

Antes: $g(\mathbf{x}) = \sum_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + w_0$ Ahora: $g(\mathbf{x}) = \sum_i \alpha_i \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle + w_0$.

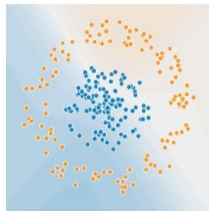
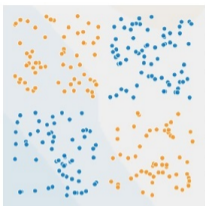
Si definimos un kernel $K_\Phi(\mathbf{u}, \mathbf{v}) := \langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle$, entonces

$$g(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + w_0.$$

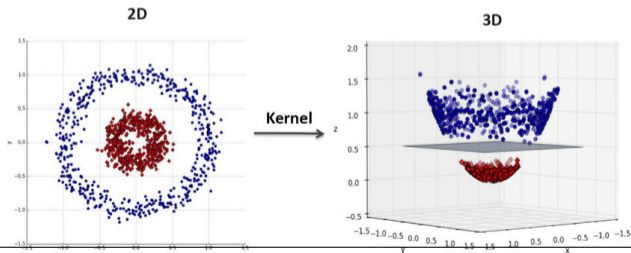
Ejemplos:

- kernel polinomial $K_\Phi(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^p$.
- kernel base radial gaussiano $K_\Phi(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$.
- Otros: lineal, sigmoide, otras RBFs, ...

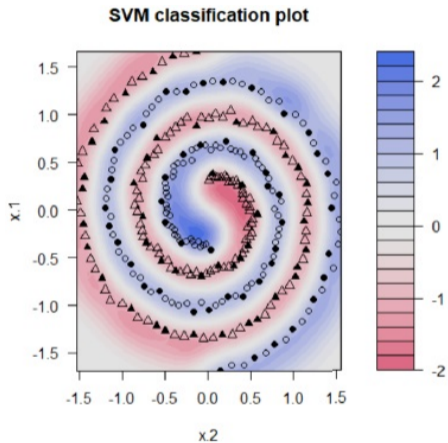
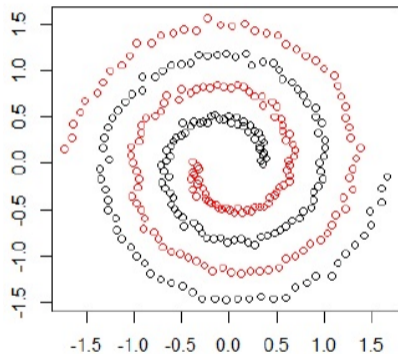
Caso general



Un kernel polinomial de grado 2 puede separar los datos.



Caso general



Un kernel RBF puede separar los datos en espiral.