Clasificación
Aprendizaje de Máquinas

Felipe Tobar

CMM - Universidad de Chile

9 de marzo de 2021

Contenido

Problema de Clasificación

Introducción - Problema de Clasificación

El problema de clasificación dice relación con la identificación del conjunto, categoría o *clase* a la cual pertenece un elemento en base a sus *características*.

Introducción - Problema de Clasificación

El problema de clasificación dice relación con la identificación del conjunto, categoría o *clase* a la cual pertenece un elemento en base a sus *características*.

En el contexto del aprendizaje supervisado, puede ser visto como un problema de regresión.

Introducción - Problema de Clasificación

El problema de clasificación dice relación con la identificación del conjunto, categoría o *clase* a la cual pertenece un elemento en base a sus *características*.

En el contexto del aprendizaje supervisado, puede ser visto como un problema de regresión.

En efecto, basta suponer que y variable de salida (o variable dependiente) es *categórica* y usualmente denotada por $\{0, 1\}$ en el caso binario o para el caso multiclase $\{1 \dots K\}$

Introducción - Problema de Clasificación

El problema de clasificación dice relación con la identificación del conjunto, categoría o *clase* a la cual pertenece un elemento en base a sus *características*.

En el contexto del aprendizaje supervisado, puede ser visto como un problema de regresión.

En efecto, basta suponer que y variable de salida (o variable dependiente) es *categórica* y usualmente denotada por $\{0, 1\}$ en el caso binario o para el caso multiclase $\{1 \dots K\}$

Entre los métodos de clasificación existentes, trabajaremos en

1. k vecinos más cercanos

Introducción - Problema de Clasificación

El problema de clasificación dice relación con la identificación del conjunto, categoría o *clase* a la cual pertenece un elemento en base a sus *características*.

En el contexto del aprendizaje supervisado, puede ser visto como un problema de regresión.

En efecto, basta suponer que y variable de salida (o variable dependiente) es *categórica* y usualmente denotada por $\{0, 1\}$ en el caso binario o para el caso multiclase $\{1 \dots K\}$

Entre los métodos de clasificación existentes, trabajaremos en

1. k vecinos más cercanos
2. Regresión Logística

Introducción - Problema de Clasificación

El problema de clasificación dice relación con la identificación del conjunto, categoría o *clase* a la cual pertenece un elemento en base a sus *características*.

En el contexto del aprendizaje supervisado, puede ser visto como un problema de regresión.

En efecto, basta suponer que y variable de salida (o variable dependiente) es *categórica* y usualmente denotada por $\{0, 1\}$ en el caso binario o para el caso multiclase $\{1 \dots K\}$

Entre los métodos de clasificación existentes, trabajaremos en

1. k vecinos más cercanos
2. Regresión Logística
3. Support Vector Machines (SVM)

Introducción - Problema de Clasificación

El problema de clasificación dice relación con la identificación del conjunto, categoría o *clase* a la cual pertenece un elemento en base a sus *características*.

En el contexto del aprendizaje supervisado, puede ser visto como un problema de regresión.

En efecto, basta suponer que y variable de salida (o variable dependiente) es *categórica* y usualmente denotada por $\{0, 1\}$ en el caso binario o para el caso multiclase $\{1 \dots K\}$

Entre los métodos de clasificación existentes, trabajaremos en

1. k vecinos más cercanos
2. Regresión Logística
3. Support Vector Machines (SVM)

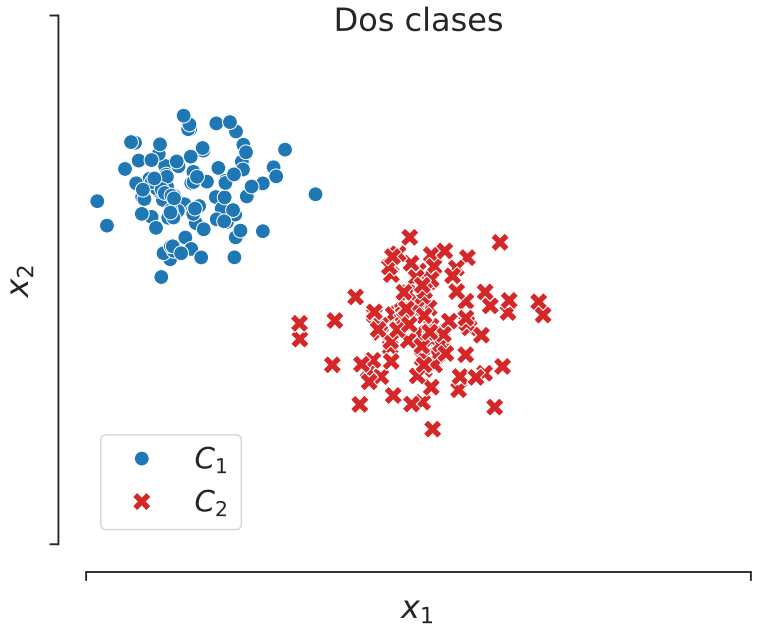

Fig. 1. Ejemplo del problema de clasificación binaria, donde la clase $\mathcal{C}_1$ está presentada en azul y la clase $\mathcal{C}_2$ en rojo.

Clasificación Lineal

Clasificación Lineal - Binaria

Consideremos el caso binario K=2 clases, proponemos un modelo lineal para relacionar la variable independiente con su clase, es decir, $y(x) = a^T x + b$ tal que $x \in \mathcal{C}_1$ si $y(x) \geq 0$, en caso contrario, $x \in \mathcal{C}_2$.

Clasificación Lineal - Binaria

Consideremos el caso binario K=2 clases, proponemos un modelo lineal para relacionar la variable independiente con su clase, es decir, $y(x) = a^T x + b$ tal que $x \in \mathcal{C}_1$ si $y(x) \geq 0$, en caso contrario, $x \in \mathcal{C}_2$.
Para encontrar los parámetros a, b óptimos, sean x_1 y x_2 en la región de decisión $y(x) = 0$

Clasificación Lineal - Binaria

Consideremos el caso binario K=2 clases, proponemos un modelo lineal para relacionar la variable independiente con su clase, es decir, $y(x) = a^T x + b$ tal que $x \in \mathcal{C}_1$ si $y(x) \geq 0$, en caso contrario, $x \in \mathcal{C}_2$.
Para encontrar los parámetros a, b óptimos, sean x_1 y x_2 en la región de decisión $y(x) = 0$

$$\begin{aligned} 0 &= y(x_1) - y(x_2) \\ &= a^\top x_1 + b - a^\top x_2 - b \\ &= a^\top (x_1 - x_2). \end{aligned}$$

Clasificación Lineal - Binaria

Consideremos el caso binario K=2 clases, proponemos un modelo lineal para relacionar la variable independiente con su clase, es decir, $y(x) = a^T x + b$ tal que $x \in \mathcal{C}_1$ si $y(x) \geq 0$, en caso contrario, $x \in \mathcal{C}_2$.
Para encontrar los parámetros a, b óptimos, sean x_1 y x_2 en la región de decisión $y(x) = 0$

$$\begin{aligned} 0 &= y(x_1) - y(x_2) \\ &= a^\top x_1 + b - a^\top x_2 - b \\ &= a^\top (x_1 - x_2). \end{aligned}$$

Además, observemos que para cualquier x en la región de decisión se tiene que

$$\|\text{proy}_a(x)\| = \|x\| \, cos(\theta) = \|x\| \, \frac{a^\top x}{\|a\| \cdot \|x\|} = -\frac{b}{\|a\|}$$

Con lo que tenemos una intepretación geométrica de ambos parámetros.

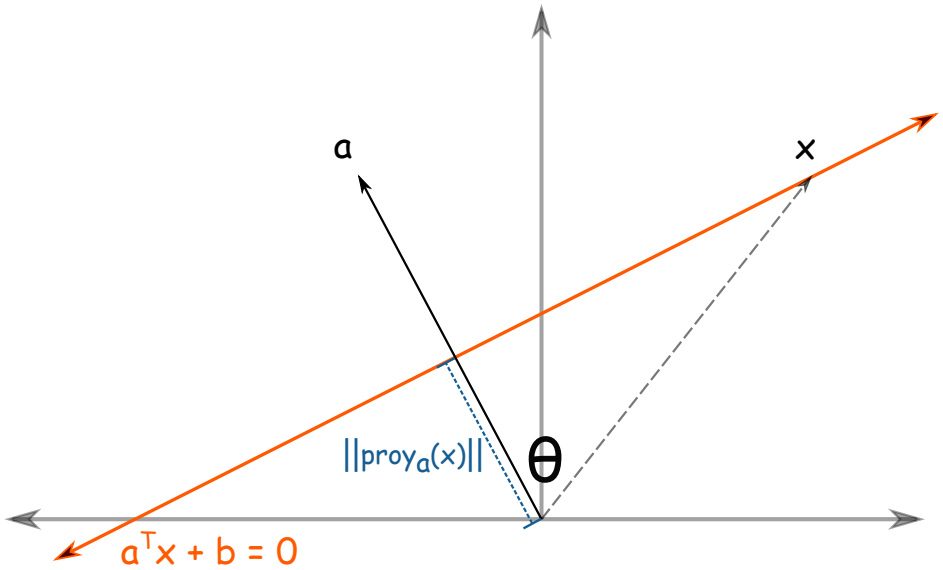

Fig. 2. Proyección de un punto sobre la región de decisión.

Clasificación Lineal - Binaria

También es posible interpretar $y(x)$ como una distancia con signo entre un $x \in \mathbb{R}^M$ cualquiera y la superficie de decisión.

También es posible interpretar $y(x)$ como una distancia con signo entre un $x \in \mathbb{R}^M$ cualquiera y la superficie de decisión.
Para ver esto, consideremos $x \in \mathbb{R}^M$ y descompongámoslo en dos componentes: la primera denotada por $x_\perp$, la cual es la proyección ortogonal de x en el hiperplano de decisión, y la segunda que es perpendicular al hiperplano (y consecuentemente paralela al vector a) denotada por $r\frac{a}{\|a\|}$, donde r denota la distancia (positiva o negativa) entre x y el hiperplano de decisión. Expresamos entonces

También es posible interpretar $y(x)$ como una distancia con signo entre un $x \in \mathbb{R}^M$ cualquiera y la superficie de decisión.
Para ver esto, consideremos $x \in \mathbb{R}^M$ y descompongámoslo en dos componentes: la primera denotada por $x_\perp$, la cual es la proyección ortogonal de x en el hiperplano de decisión, y la segunda que es perpendicular al hiperplano (y consecuentemente paralela al vector a) denotada por $r\frac{a}{\|a\|}$, donde r denota la distancia (positiva o negativa) entre x y el hiperplano de decisión. Expresamos entonces

$$x = x_\perp + r\frac{a}{\|a\|},$$

y observamos que

También es posible interpretar $y(x)$ como una distancia con signo entre un $x \in \mathbb{R}^M$ cualquiera y la superficie de decisión.
Para ver esto, consideremos $x \in \mathbb{R}^M$ y descompongámoslo en dos componentes: la primera denotada por $x_\perp$, la cual es la proyección ortogonal de x en el hiperplano de decisión, y la segunda que es perpendicular al hiperplano (y consecuentemente paralela al vector a) denotada por $r\frac{a}{\|a\|}$, donde r denota la distancia (positiva o negativa) entre x y el hiperplano de decisión. Expresamos entonces

$$x = x_\perp + r\frac{a}{\|a\|},$$

y observamos que

$$y(x) = a^\top x + b = a^\top \left(x_\perp + r\frac{a}{\|a\|} \right) + b = \underbrace{a^\top x_\perp + b}_{=0} + r\frac{a^\top a}{\|a\|} = r||a||.$$

También es posible interpretar $y(x)$ como una distancia con signo entre un $x \in \mathbb{R}^M$ cualquiera y la superficie de decisión.
Para ver esto, consideremos $x \in \mathbb{R}^M$ y descompongámoslo en dos componentes: la primera denotada por $x_\perp$, la cual es la proyección ortogonal de x en el hiperplano de decisión, y la segunda que es perpendicular al hiperplano (y consecuentemente paralela al vector a) denotada por $r\frac{a}{\|a\|}$, donde r denota la distancia (positiva o negativa) entre x y el hiperplano de decisión. Expresamos entonces

$$x = x_\perp + r\frac{a}{\|a\|},$$

y observamos que

$$y(x) = a^\top x + b = a^\top \left(x_\perp + r\frac{a}{\|a\|} \right) + b = \underbrace{a^\top x_\perp + b}_{=0} + r\frac{a^\top a}{\|a\|} = r||a||.$$

Luego $r = \frac{y(x)}{\|a\|}$ y como r es una medida con signo, $y(x)$ también lo es.

Clasificación Lineal - Múltiples clases

El caso de múltiples clases ($K > 2$) puede ser enfrentado mediante una extensión del caso binario, algunas de ellas

1. **One versus rest:** Construcción de $K - 1$ clasificadores binarios que discrimina una clase $\mathcal{C}_k$ del resto

Clasificación Lineal - Múltiples clases

El caso de múltiples clases ($K > 2$) puede ser enfrentado mediante una extensión del caso binario, algunas de ellas

1. **One versus rest:** Construcción de $K - 1$ clasificadores binarios que discrimina una clase $\mathcal{C}_k$ del resto
2. **One versus one:** Construccion de $K(K-1)/2$ clasificadores binarios que discriminan entre cada par posible de clases

Clasificación Lineal - Múltiples clases

El caso de múltiples clases ($K > 2$) puede ser enfrentado mediante una extensión del caso binario, algunas de ellas

1. **One versus rest:** Construcción de $K - 1$ clasificadores binarios que discrimina una clase $\mathcal{C}_k$ del resto
2. **One versus one:** Construccion de $K(K-1)/2$ clasificadores binarios que discriminan entre cada par posible de clases

¿Qué problema presentan estos métodos?

Clasificación Lineal - Múltiples clases

El caso de múltiples clases ($K > 2$) puede ser enfrentado mediante una extensión del caso binario, algunas de ellas

1. **One versus rest:** Construcción de $K - 1$ clasificadores binarios que discrimina una clase $\mathcal{C}_k$ del resto
2. **One versus one:** Construccion de $K(K-1)/2$ clasificadores binarios que discriminan entre cada par posible de clases

¿Qué problema presentan estos métodos? Busquemos otra forma

Clasificación Lineal - Múltiples clases

El caso de múltiples clases ($K > 2$) puede ser enfrentado mediante una extensión del caso binario, algunas de ellas

1. **One versus rest:** Construcción de $K - 1$ clasificadores binarios que discrimina una clase $\mathcal{C}_k$ del resto
2. **One versus one:** Construccion de $K(K-1)/2$ clasificadores binarios que discriminan entre cada par posible de clases

¿Qué problema presentan estos métodos? Busquemos otra forma

Una alternativa más robusta para resolver el problema de clasificación multiclase es construir un clasificador para K clases que contiene K funciones lineales de la forma

$$y_k(x) = a_k^\top x + b_k, \quad k = 1, \ldots, K.$$

Clasificación Lineal - Múltiples clases

El caso de múltiples clases ($K > 2$) puede ser enfrentado mediante una extensión del caso binario, algunas de ellas

1. **One versus rest:** Construcción de $K - 1$ clasificadores binarios que discrimina una clase $\mathcal{C}_k$ del resto
2. **One versus one:** Construccion de $K(K-1)/2$ clasificadores binarios que discriminan entre cada par posible de clases

¿Qué problema presentan estos métodos? Busquemos otra forma

Una alternativa más robusta para resolver el problema de clasificación multiclase es construir un clasificador para K clases que contiene K funciones lineales de la forma

$$y_k(x) = a_k^\top x + b_k, \quad k = 1, \ldots, K.$$

Donde x es asignado a la clase $\mathcal{C}_k$ si y solo si $y_k(x) > y_j(x), \forall j \neq k$, es decir:

$$\mathcal{C}(x) = \arg\max_k y_j(x).$$

Clasificación Lineal - Múltiples clases

El caso de múltiples clases ($K > 2$) puede ser enfrentado mediante una extensión del caso binario, algunas de ellas

1. **One versus rest:** Construcción de $K - 1$ clasificadores binarios que discrimina una clase $\mathcal{C}_k$ del resto
2. **One versus one:** Construccion de $K(K-1)/2$ clasificadores binarios que discriminan entre cada par posible de clases

¿Qué problema presentan estos métodos? Busquemos otra forma

Una alternativa más robusta para resolver el problema de clasificación multiclase es construir un clasificador para K clases que contiene K funciones lineales de la forma

$$y_k(x) = a_k^\top x + b_k, \quad k = 1, \ldots, K.$$

Donde x es asignado a la clase $\mathcal{C}_k$ si y solo si $y_k(x) > y_j(x), \forall j \neq k$, es decir:

$$\mathcal{C}(x) = \arg\max_k y_j(x).$$

¿Qué ventajas posee este método?

Ajuste mediante mínimos cuadrados

Ajuste mediante mínimos cuadrados

Ya hemos planteado el modelo y analizado el rol y significado de cada uno de sus parámetros; ahora queda por estudiar cómo determinar dichos parámetros a y b, dado un conjunto de datos $\mathcal{D}$.

Ajuste mediante mínimos cuadrados

Ya hemos planteado el modelo y analizado el rol y significado de cada uno de sus parámetros; ahora queda por estudiar cómo determinar dichos parámetros a y b, dado un conjunto de datos $\mathcal{D}$.
Consideremos el punto $x \in \mathbb{R}^M$ con clase $c \in \{\mathcal{C}_k\}_{k=1}^K$. Usaremos la *codificación* $t \in \{0,1\}^K$ para representar la pertenencia de x a su respectiva clase. Es decir,

Ajuste mediante mínimos cuadrados

Ya hemos planteado el modelo y analizado el rol y significado de cada uno de sus parámetros; ahora queda por estudiar cómo determinar dichos parámetros a y b, dado un conjunto de datos $\mathcal{D}$.
Consideremos el punto $x \in \mathbb{R}^M$ con clase $c \in \{\mathcal{C}_k\}_{k=1}^K$. Usaremos la *codificación* $t \in \{0,1\}^K$ para representar la pertenencia de x a su respectiva clase. Es decir,

$$c = \mathcal{C}_j \Leftrightarrow [t]_j = 1 \wedge [t]_i = 0, \quad \forall i \neq j.$$

Este tipo de codificación es conocida como *one-hot encoding*. **¿Por qué la usamos?**

Ajuste mediante mínimos cuadrados

Ya hemos planteado el modelo y analizado el rol y significado de cada uno de sus parámetros; ahora queda por estudiar cómo determinar dichos parámetros a y b, dado un conjunto de datos $\mathcal{D}$.
Consideremos el punto $x \in \mathbb{R}^M$ con clase $c \in \{\mathcal{C}_k\}_{k=1}^K$. Usaremos la *codificación* $t \in \{0,1\}^K$ para representar la pertenencia de x a su respectiva clase. Es decir,

$$c = \mathcal{C}_j \Leftrightarrow [t]_j = 1 \wedge [t]_i = 0, \quad \forall i \neq j.$$

Este tipo de codificación es conocida como *one-hot encoding*. **¿Por qué la usamos?**
Asumiendo entonces un modelo lineal para cada clase $\mathcal{C}_k$, se tiene que

$$y_k(x) = a_k^\top x + b_k = \tilde{\theta}_k^\top \tilde{x}, \quad \text{donde } \tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix} \in \mathbb{R}^{M+1}, \quad \tilde{\theta}_k = \begin{pmatrix} a_k \\ b_k \end{pmatrix} \in \mathbb{R}^{M+1}$$

Ajuste mediante mínimos cuadrados

Ya hemos planteado el modelo y analizado el rol y significado de cada uno de sus parámetros; ahora queda por estudiar cómo determinar dichos parámetros a y b, dado un conjunto de datos $\mathcal{D}$.
Consideremos el punto $x \in \mathbb{R}^M$ con clase $c \in \{\mathcal{C}_k\}_{k=1}^K$. Usaremos la *codificación* $t \in \{0,1\}^K$ para representar la pertenencia de x a su respectiva clase. Es decir,

$$c = \mathcal{C}_j \Leftrightarrow [t]_j = 1 \wedge [t]_i = 0, \quad \forall i \neq j.$$

Este tipo de codificación es conocida como *one-hot encoding*. **¿Por qué la usamos?**
Asumiendo entonces un modelo lineal para cada clase $\mathcal{C}_k$, se tiene que

$$y_k(x) = a_k^\top x + b_k = \tilde{\theta}_k^\top \tilde{x}, \quad \text{donde } \tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix} \in \mathbb{R}^{M+1}, \quad \tilde{\theta}_k = \begin{pmatrix} a_k \\ b_k \end{pmatrix} \in \mathbb{R}^{M+1}$$

Lo anterior se puede unir en un único sistema matricial:

$$\tilde{\Theta} = (\theta_1, \cdots, \theta_K) \in \mathbb{R}^{(M+1)\times K} \implies y(x) = \begin{pmatrix} y_1(x) \\ \vdots \\ y_K(x) \end{pmatrix} = \tilde{\Theta}^\top \tilde{x}.$$

Ajuste mediante mínimos cuadrados

Con la notación establecida, ahora podemos enfocarnos en el entrenamiento del modelo. Para esto consideremos un conjunto de entrenamiento $\{(x_n, t_n)\}_{n=1}^{N}$. El enfoque de entrenamiento será el correspondiente a mínimos cuadrados asociado al error de asignación:

$$J = \sum_{i=1}^{N} \left\| t_i - \tilde{\Theta}^{\top} \tilde{x}_i \right\|_2^2$$

Ajuste mediante mínimos cuadrados

Con la notación establecida, ahora podemos enfocarnos en el entrenamiento del modelo. Para esto consideremos un conjunto de entrenamiento $\{(x_n, t_n)\}_{n=1}^{N}$. El enfoque de entrenamiento será el correspondiente a mínimos cuadrados asociado al error de asignación:

$$J = \sum_{i=1}^{N} \left\| t_i - \tilde{\Theta}^\top \tilde{x}_i \right\|_2^2$$

Por otra parte, definiendo las siguientes matrices:

$$T = \begin{pmatrix} t_1^\top \\ \vdots \\ t_N^\top \end{pmatrix} \in \mathbb{R}^{N \times K}, \qquad \tilde{X} = \begin{pmatrix} t_1^\top \\ \vdots \\ t_N^\top \end{pmatrix} \in \mathbb{R}^{N \times (M+1)}$$

Ajuste mediante mínimos cuadrados

Con la notación establecida, ahora podemos enfocarnos en el entrenamiento del modelo. Para esto consideremos un conjunto de entrenamiento $\{(x_n, t_n)\}_{n=1}^{N}$. El enfoque de entrenamiento será el correspondiente a mínimos cuadrados asociado al error de asignación:

$$J = \sum_{i=1}^{N} \left\| t_i - \tilde{\Theta}^\top \tilde{x}_i \right\|_2^2$$

Por otra parte, definiendo las siguientes matrices:

$$T = \begin{pmatrix} t_1^\top \\ \vdots \\ t_N^\top \end{pmatrix} \in \mathbb{R}^{N \times K}, \qquad \tilde{X} = \begin{pmatrix} t_1^\top \\ \vdots \\ t_N^\top \end{pmatrix} \in \mathbb{R}^{N \times (M+1)}$$

se tiene el siguiente resultado:

Ajuste mediante mínimos cuadrados

Lemma

Bajo la notación anterior, $J = \mathrm{Tr}\left((\tilde{X}\tilde{\Theta} - T)^\top (\tilde{X}\tilde{\Theta} - T)\right)$ *y su mínimo es alcanzado en:*

$$\tilde{\Theta} = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top T$$

donde Tr *corresponde al operador traza:* $A \in \mathbb{R}^{n \times n} \mapsto \mathrm{Tr}(A) := \sum_{i=1}^{n} a_{ii}$.

Demostración.

$$J = \sum_{i=1}^{N} \left\| t_i - \tilde{\Theta}^\top \tilde{x}_i \right\|_2^2 = \sum_{i=1}^{N} \left\| \left(T - \tilde{X}\tilde{\Theta} \right)_{i\cdot} \right\|_2^2 = \sum_{i=1}^{N} \sum_{j=1}^{K} \left(T - \tilde{X}\tilde{\Theta} \right)_{ij} \left(T - \tilde{X}\tilde{\Theta} \right)_{ij}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{K} \left(T - \tilde{X}\tilde{\Theta} \right)_{ji}^{\top} \left(T - \tilde{X}\tilde{\Theta} \right)_{ij} = \sum_{j=1}^{K} \left[\left(T - \tilde{X}\tilde{\Theta} \right)^{\top} \left(T - \tilde{X}\tilde{\Theta} \right) \right]_{jj}$$

$$= \mathrm{Tr}\left((\tilde{X}\tilde{\Theta} - T)^\top (\tilde{X}\tilde{\Theta} - T) \right).$$

Por otra parte:

$$\frac{\partial J}{\partial \tilde{\Theta}} = 2(\tilde{X}\tilde{\Theta} - T)^\top \tilde{X} = 0 \iff \tilde{\Theta}^\top \tilde{X}^\top \tilde{X} - T^\top \tilde{X} = 0$$

$$\iff \tilde{\Theta}^\top = T^\top \tilde{X} (\tilde{X}^\top \tilde{X})^{-1} \iff \tilde{\Theta} = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top T$$

Y dado que J es estrictamente convexo, su mínimo se alcanza en su único punto crítico.

□

Ajuste mediante mínimos cuadrados

Problemáticas conceptuales de este enfoque:

1. Sensibilidad a presencia de puntos aislados (outliers)

Ajuste mediante mínimos cuadrados

Problemáticas conceptuales de este enfoque:

1. Sensibilidad a presencia de puntos aislados (outliers)
2. Intervención "manual" de las etiquetas

Ajuste mediante mínimos cuadrados

Problemáticas conceptuales de este enfoque:

1. Sensibilidad a presencia de puntos aislados (outliers)
2. Intervención "manual" de las etiquetas

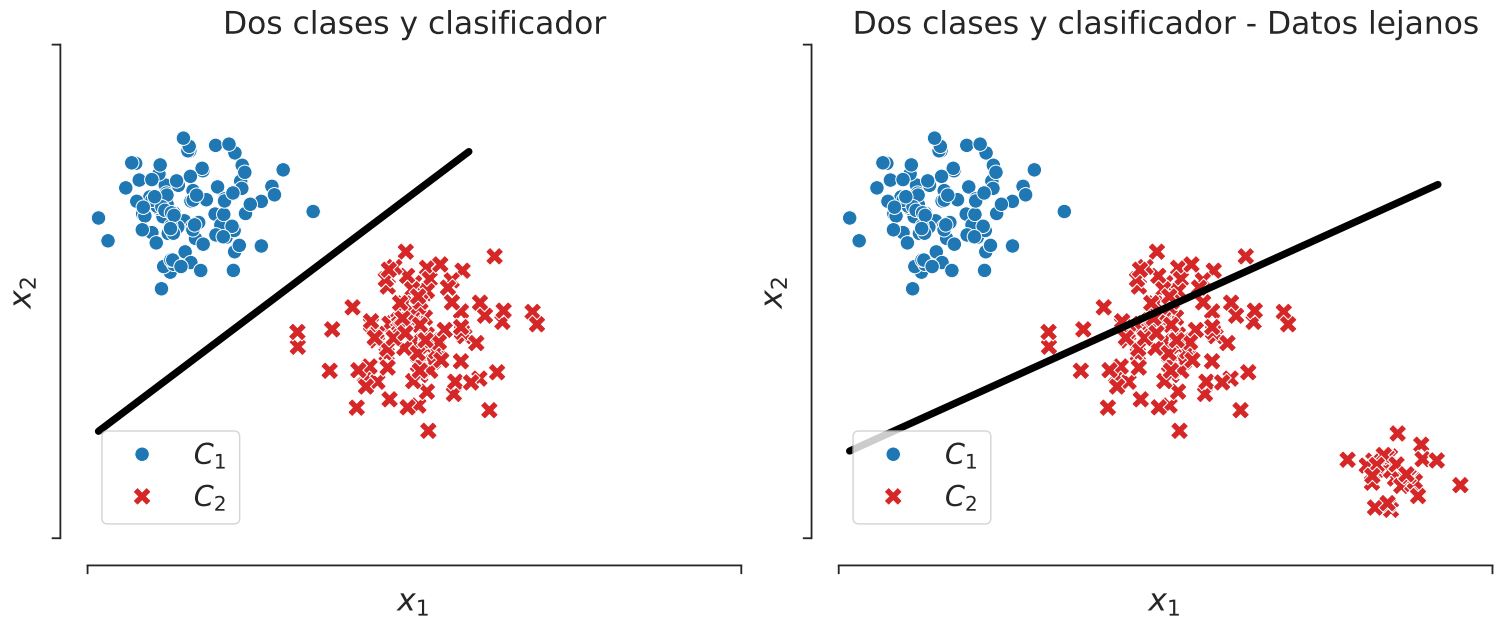

Fig. 3. Ejemplo ilustrativo sobre cómo los puntos lejanos de una clase pueden afectar incorrectamente los resultados.

El perceptrón

El perceptrón - Introducción

Las nociones básicas que hemos visto hasta ahora para lidiar con el problema de clasificación tienen dos problemas conceptuales.

El perceptrón - Introducción

Las nociones básicas que hemos visto hasta ahora para lidiar con el problema de clasificación tienen dos problemas conceptuales.

1. Falta de una métrica correcta

El perceptrón - Introducción

Las nociones básicas que hemos visto hasta ahora para lidiar con el problema de clasificación tienen dos problemas conceptuales.

1. Falta de una métrica correcta
2. No existe una *función de verosimilitud* apropiada

El perceptrón - Introducción

Las nociones básicas que hemos visto hasta ahora para lidiar con el problema de clasificación tienen dos problemas conceptuales.

1. Falta de una métrica correcta
2. No existe una *función de verosimilitud* apropiada

La incorporación de esta función que conecta el modelo lineal con la clase, resulta en un *modelo lineal generalizado*, es decir, una modelo lineal conectado a una función no-lineal que llamaremos *función de enlace*.

El perceptrón - Introducción

Las nociones básicas que hemos visto hasta ahora para lidiar con el problema de clasificación tienen dos problemas conceptuales.

1. Falta de una métrica correcta
2. No existe una *función de verosimilitud* apropiada

La incorporación de esta función que conecta el modelo lineal con la clase, resulta en un *modelo lineal generalizado*, es decir, una modelo lineal conectado a una función no-lineal que llamaremos *función de enlace*.
Sin embargo, el desafío más importante en esta construcción es que el modelo resultante ya no es lineal, ni en la entrada ni en los parámetros , pues una verosimilitud (función de enlace) lineal nunca nos llevará de un espacio de inputs (hemos asumido $\mathbb{R}^M$) al espacio de categorías $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$, es decir, necesitamos una no-linealidad "después" de la parte lineal

El perceptrón - Introducción

Las nociones básicas que hemos visto hasta ahora para lidiar con el problema de clasificación tienen dos problemas conceptuales.

1. Falta de una métrica correcta
2. No existe una *función de verosimilitud* apropiada

La incorporación de esta función que conecta el modelo lineal con la clase, resulta en un *modelo lineal generalizado*, es decir, una modelo lineal conectado a una función no-lineal que llamaremos *función de enlace*.
Sin embargo, el desafío más importante en esta construcción es que el modelo resultante ya no es lineal, ni en la entrada ni en los parámetros , pues una verosimilitud (función de enlace) lineal nunca nos llevará de un espacio de inputs (hemos asumido $\mathbb{R}^M$) al espacio de categorías $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$, es decir, necesitamos una no-linealidad "después" de la parte lineal
Una forma de resolver estas problemáticas es mediante el uso del *Perceptrón* (Rosenblatt, 1958), un modelo de clasificación binario que tuvo mucha importancia en el área de reconocimiento de patrones.

El perceptrón

El Perceptrón consiste en una función no lineal fija usada para transformar x en un vector de características[1] $\phi(x) \in \mathbb{R}^D$, que luego es usado para generar un modelo lineal *generalizado* con función de enlace no lineal $f(\cdot)$ de la siguiente forma:

$$y(x) = f(\theta^\top \phi(x))$$

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

[1] En este caso consideramos no linealidad antes y después de la parte lineal, sin embargo, considerar la entrada como x o como $\phi(x)$ es equivalente en base a lo visto en los modelos lineales en los parámetros.

El perceptrón

El Perceptrón consiste en una función no lineal fija usada para transformar x en un vector de características[1] $\phi(x) \in \mathbb{R}^D$, que luego es usado para generar un modelo lineal *generalizado* con función de enlace no lineal $f(\cdot)$ de la siguiente forma:

$$y(x) = f(\theta^\top \phi(x))$$

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

El Perceptrón entonces asigna x a la clase $\mathcal{C}_1$ si $y(x) = +1$ y asignará x a la clase $\mathcal{C}_2$ cuando $y(x) = -1$. Notemos que para el caso que ϕ es lineal, este es el mismo clasificador presentado en la Sección de clasificación lineal, pero en este caso el criterio para asignar la clase es **parte del modelo**.

[1] En este caso consideramos no linealidad antes y después de la parte lineal, sin embargo, considerar la entrada como x o como $\phi(x)$ es equivalente en base a lo visto en los modelos lineales en los parámetros.

El perceptrón

El Perceptrón consiste en una función no lineal fija usada para transformar x en un vector de características[1] $\phi(x) \in \mathbb{R}^D$, que luego es usado para generar un modelo lineal *generalizado* con función de enlace no lineal $f(\cdot)$ de la siguiente forma:

$$y(x) = f(\theta^\top \phi(x))$$

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

El Perceptrón entonces asigna x a la clase $\mathcal{C}_1$ si $y(x) = +1$ y asignará x a la clase $\mathcal{C}_2$ cuando $y(x) = -1$. Notemos que para el caso que ϕ es lineal, este es el mismo clasificador presentado en la Sección de clasificación lineal, pero en este caso el criterio para asignar la clase es **parte del modelo**.
Usando el hecho que las etiquetas están representadas por la codificación $t \in \{1, -1\}$, la condición de asignación puede ser cubiertas por la expresión:

$$\theta^\top \phi(x_n) t_n > 0, \quad \forall (x_n, t_n) \in \mathcal{D}.$$

[1] En este caso consideramos no linealidad antes y después de la parte lineal, sin embargo, considerar la entrada como x o como $\phi(x)$ es equivalente en base a lo visto en los modelos lineales en los parámetros.

El perceptrón

Podemos entonces satisfacer esta restricción mediante el "criterio del perceptrón", el cual se basa en examinar los elementos de $\mathcal{D}$ que fueron clasificados incorrectamente. Este criterio asocia a los puntos clasificados correctamente error 0 y a los puntos mal clasificados error $-\theta^\top \phi(x)t > 0$. De esta forma, si denotamos $\mathcal{M}$ el conjunto de puntos mal clasificados, se debe minimizar la siguiente función objetivo:

El perceptrón

Podemos entonces satisfacer esta restricción mediante el "criterio del perceptrón", el cual se basa en examinar los elementos de $\mathcal{D}$ que fueron clasificados incorrectamente. Este criterio asocia a los puntos clasificados correctamente error 0 y a los puntos mal clasificados error $-\theta^\top \phi(x)t > 0$. De esta forma, si denotamos $\mathcal{M}$ el conjunto de puntos mal clasificados, se debe minimizar la siguiente función objetivo:

$$J_{\mathsf{P}}(\theta, x) = \mathbb{E}\left(-\theta^\top \phi(x)t(x)\mathbb{1}_{\theta^\top \phi(x)t(x)\leq 0}\right)$$

El perceptrón

Podemos entonces satisfacer esta restricción mediante el "criterio del perceptrón", el cual se basa en examinar los elementos de $\mathcal{D}$ que fueron clasificados incorrectamente. Este criterio asocia a los puntos clasificados correctamente error 0 y a los puntos mal clasificados error $-\theta^\top \phi(x)t > 0$. De esta forma, si denotamos $\mathcal{M}$ el conjunto de puntos mal clasificados, se debe minimizar la siguiente función objetivo:

$$J_{\mathrm{P}}(\theta, x) = \mathbb{E}\left(-\theta^\top \phi(x)t(x)\mathbb{1}_{\theta^\top \phi(x)t(x)\leq 0}\right)$$

$$\approx -\sum_{(x_i,t_i)\in\mathcal{D}} \theta^\top \phi(x_i)t_i\mathbb{1}_{\theta^\top \phi(x_i)t_i\leq 0} = -\sum_{(x_i,t_i)\in\mathcal{M}} \theta^\top \phi(x_i)t_i$$

Para el problema de minimización del funcional del perceptrón, se puede utilizar el método del gradiente estocástico (ver anexo).

El perceptrón

En este caso, el algoritmo iterativo tiene la siguiente estructura:

El perceptrón

En este caso, el algoritmo iterativo tiene la siguiente estructura:

$$\begin{aligned}\theta^{\tau+1} &= \theta^{\tau} - \eta_{\tau} \nabla_{\theta} J_{\mathrm{P}}(\theta^{\tau}, x_i) \\ &= \theta^{\tau} + \eta_{\tau} \phi(x_i) t_i .\end{aligned}$$

El perceptrón

En este caso, el algoritmo iterativo tiene la siguiente estructura:

$$\begin{aligned}\theta^{\tau+1} &= \theta^{\tau} - \eta_{\tau} \nabla_{\theta} J_{\mathrm{P}}(\theta^{\tau}, x_i) \\ &= \theta^{\tau} + \eta_{\tau} \phi(x_i) t_i.\end{aligned}$$

Es importante notar que al actualizar el vector θ, el conjunto de puntos mal clasificados $\mathcal{M}$ va a cambiar, pues (esperamos que) en cada iteración los elementos del conjunto de puntos mal clasificados vaya disminuyendo.
Por lo tanto, el algoritmo de entrenamiento para el perceptrón es el siguiente:

El perceptrón

En este caso, el algoritmo iterativo tiene la siguiente estructura:

$$\begin{aligned}\theta^{\tau+1} &= \theta^{\tau} - \eta_{\tau} \nabla_{\theta} J_{\mathrm{P}}(\theta^{\tau}, x_i) \\ &= \theta^{\tau} + \eta_{\tau} \phi(x_i) t_i .\end{aligned}$$

Es importante notar que al actualizar el vector θ, el conjunto de puntos mal clasificados $\mathcal{M}$ va a cambiar, pues (esperamos que) en cada iteración los elementos del conjunto de puntos mal clasificados vaya disminuyendo.
Por lo tanto, el algoritmo de entrenamiento para el perceptrón es el siguiente:

i) se recorre el conjunto de puntos de entrenamiento $\{x_i\}_{i=1}^{N}$,

El perceptrón

En este caso, el algoritmo iterativo tiene la siguiente estructura:

$$\begin{aligned}\theta^{\tau+1} &= \theta^{\tau} - \eta_{\tau} \nabla_{\theta} J_{\mathrm{P}}(\theta^{\tau}, x_i) \\ &= \theta^{\tau} + \eta_{\tau} \phi(x_i) t_i .\end{aligned}$$

Es importante notar que al actualizar el vector θ, el conjunto de puntos mal clasificados $\mathcal{M}$ va a cambiar, pues (esperamos que) en cada iteración los elementos del conjunto de puntos mal clasificados vaya disminuyendo.
Por lo tanto, el algoritmo de entrenamiento para el perceptrón es el siguiente:

i) se recorre el conjunto de puntos de entrenamiento $\{x_i\}_{i=1}^{N}$,

ii) si el punto x_i fue clasificado correctamente el vector de pesos de mantiene igual

El perceptrón

En este caso, el algoritmo iterativo tiene la siguiente estructura:

$$\begin{aligned}\theta^{\tau+1} &= \theta^{\tau} - \eta_{\tau} \nabla_{\theta} J_{\mathrm{P}}(\theta^{\tau}, x_i) \\ &= \theta^{\tau} + \eta_{\tau} \phi(x_i) t_i.\end{aligned}$$

Es importante notar que al actualizar el vector θ, el conjunto de puntos mal clasificados $\mathcal{M}$ va a cambiar, pues (esperamos que) en cada iteración los elementos del conjunto de puntos mal clasificados vaya disminuyendo.
Por lo tanto, el algoritmo de entrenamiento para el perceptrón es el siguiente:

i) se recorre el conjunto de puntos de entrenamiento $\{x_i\}_{i=1}^{N}$,

ii) si el punto x_i fue clasificado correctamente el vector de pesos de mantiene igual

iii) si x_i fue clasificado incorrectamente, el vector θ^{τ} es actualizado según la ecuación anterior con $\eta = 1$ mediante

$$\theta^{\tau+1} = \theta^{\tau} + \phi(x_i) t_i.$$

El perceptrón

En este caso, el algoritmo iterativo tiene la siguiente estructura:

$$\begin{aligned}\theta^{\tau+1} &= \theta^{\tau} - \eta_{\tau} \nabla_{\theta} J_{\mathrm{P}}(\theta^{\tau}, x_i) \\ &= \theta^{\tau} + \eta_{\tau} \phi(x_i) t_i.\end{aligned}$$

Es importante notar que al actualizar el vector θ, el conjunto de puntos mal clasificados $\mathcal{M}$ va a cambiar, pues (esperamos que) en cada iteración los elementos del conjunto de puntos mal clasificados vaya disminuyendo.
Por lo tanto, el algoritmo de entrenamiento para el perceptrón es el siguiente:

i) se recorre el conjunto de puntos de entrenamiento $\{x_i\}_{i=1}^{N}$,

ii) si el punto x_i fue clasificado correctamente el vector de pesos de mantiene igual

iii) si x_i fue clasificado incorrectamente, el vector θ^{τ} es actualizado según la ecuación anterior con $\eta = 1$ mediante

$$\theta^{\tau+1} = \theta^{\tau} + \phi(x_i) t_i.$$

Es decir, el parámetro θ está paso a paso modificado en la dirección de las características $\phi(x_i)$ con multiplicador ± 1 en base a la clase verdadera de x_i hasta que todos los puntos de $\mathcal{D}$ están bien clasificados.

Clasificación probabilística: modelo generativo

Modelo generativo

Los modelos que hemos revisado hasta este punto son del tipo *discriminativo*, es decir, modelan directamente la función $f : x \mapsto c$. Con una interpretación probabilística, esto es equivalente a modelar la probabilidad condicional $\mathbb{P}(\mathcal{C}_k|x)$, es decir, dado que conozco el input (o características de) x, cuál es la distribución de probabilidad sobre las clases. Sin embargo, hemos considerado métodos determinísticos, que solo asignan probabilidad 1 a una sola clase.

Modelo generativo

Los modelos que hemos revisado hasta este punto son del tipo *discriminativo*, es decir, modelan directamente la función $f : x \mapsto c$. Con una interpretación probabilística, esto es equivalente a modelar la probabilidad condicional $\mathbb{P}(\mathcal{C}_k|x)$, es decir, dado que conozco el input (o características de) x, cuál es la distribución de probabilidad sobre las clases. Sin embargo, hemos considerado métodos determinísticos, que solo asignan probabilidad 1 a una sola clase.

Un paradigma alternativo es considerar es un enfoque *generativo*, en el cual modelamos dos objetos: en primer lugar la "probabilidad condicional de clase" la cual representa cómo distribuyen los valores de los inputs x cuando la clase es, por ejemplo, $\mathcal{C}_k$, denotada por $\mathbb{P}(x|\mathcal{C}_k)$. En segundo lugar las "probabilidades de clase", o el prior sobre clases, denotada $\mathbb{P}(\mathcal{C}_k)$. Luego, podemos calcular la densidad posterior sobre las clases dado un input x usando el Teorema de Bayes de acuerdo a

$$\mathbb{P}(\mathcal{C}_k|x) = \frac{\mathbb{P}(x|\mathcal{C}_k)\mathbb{P}(\mathcal{C}_k)}{\mathbb{P}(x)}.$$

Modelo generativo

Para el caso de 2 clases, se tiene el siguiente desarrollo:

Modelo generativo

Para el caso de 2 clases, se tiene el siguiente desarrollo:

$$\begin{aligned}
\mathbb{P}(\mathcal{C}_1|x) &= \frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x)} \\
&= \frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1) + \mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)} \\
&= \frac{1}{1 + \frac{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}} \\
&= \frac{1}{1 + \exp(-r)} = \sigma(r).
\end{aligned}$$

Modelo generativo

Para el caso de 2 clases, se tiene el siguiente desarrollo:

$$\begin{aligned}\mathbb{P}(\mathcal{C}_1|x) &= \frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x)} \\ &= \frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1) + \mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)} \\ &= \frac{1}{1 + \frac{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}} \\ &= \frac{1}{1+\exp(-r)} = \sigma(r).\end{aligned}$$

Donde hemos introducido la notación $r = r(x) = \ln\left(\frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}\right)$ y la función logística definida mediante $\sigma(r) = \frac{1}{1+e^{-r}}$, la cual tiene propiedades que serán útiles en el entrenamiento, en particular:

Modelo generativo

Para el caso de 2 clases, se tiene el siguiente desarrollo:

$$\begin{aligned}\mathbb{P}(\mathcal{C}_1|x) &= \frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x)} \\ &= \frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1) + \mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)} \\ &= \frac{1}{1 + \frac{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}} \\ &= \frac{1}{1+\exp(-r)} = \sigma(r).\end{aligned}$$

Donde hemos introducido la notación $r = r(x) = \ln\left(\frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}\right)$ y la función logística definida mediante $\sigma(r) = \frac{1}{1+e^{-r}}$, la cual tiene propiedades que serán útiles en el entrenamiento, en particular:

$$\text{reflejo: } \sigma(-r) = 1 - \sigma(r)$$

$$\text{derivada: } \frac{d}{dr}\sigma(r) = \sigma(r)(1-\sigma(r))$$

$$\text{inversa: } r(\sigma) = \ln\left(\frac{\sigma}{1-\sigma}\right).$$

Modelo generativo

Si bien la expresión de la distribución condicional en la ecuación anterior parece una presentación antojadiza para hacer aparecer la función logística (sigmoide), pues $r = r(x)$ puede ser cualquier cosa. Sin embargo, veremos que existe una elección particular de las distribuciones condicionales de clase que lleva a un r que es efectivamente lineal en x. En general, nos referiremos a este clasificador como **regresión logística** en dicho caso, es decir, cuando $r(x) = a^\top x + b$.

Modelo generativo

Si bien la expresión de la distribución condicional en la ecuación anterior parece una presentación antojadiza para hacer aparecer la función logística (sigmoide), pues $r = r(x)$ puede ser cualquier cosa. Sin embargo, veremos que existe una elección particular de las distribuciones condicionales de clase que lleva a un r que es efectivamente lineal en x. En general, nos referiremos a este clasificador como **regresión logística** en dicho caso, es decir, cuando $r(x) = a^\top x + b$.
Podemos ahora considerar el caso de múltiples clases $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, donde un desarrollo similar al anterior resulta en:

$$\mathbb{P}(\mathcal{C}_i|x) = \frac{\mathbb{P}(x|\mathcal{C}_i)\mathbb{P}(\mathcal{C}_i)}{\sum_j \mathbb{P}(x|\mathcal{C}_j)\mathbb{P}(\mathcal{C}_j)} = \frac{\exp(s_i)}{\sum_j \exp(s_j)},$$

Modelo generativo

Si bien la expresión de la distribución condicional en la ecuación anterior parece una presentación antojadiza para hacer aparecer la función logística (sigmoide), pues $r = r(x)$ puede ser cualquier cosa. Sin embargo, veremos que existe una elección particular de las distribuciones condicionales de clase que lleva a un r que es efectivamente lineal en x. En general, nos referiremos a este clasificador como **regresión logística** en dicho caso, es decir, cuando $r(x) = a^\top x + b$.
Podemos ahora considerar el caso de múltiples clases $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, donde un desarrollo similar al anterior resulta en:

$$\mathbb{P}(\mathcal{C}_i|x) = \frac{\mathbb{P}(x|\mathcal{C}_i)\mathbb{P}(\mathcal{C}_i)}{\sum_j \mathbb{P}(x|\mathcal{C}_j)\mathbb{P}(\mathcal{C}_j)} = \frac{\exp(s_i)}{\sum_j \exp(s_j)},$$

donde hemos denotado $s_i = \log\left(\mathbb{P}(x|\mathcal{C}_i)\mathbb{P}(\mathcal{C}_i)\right)$. La función que aparece al lado derecho de la ecuación se conoce como *exponencial normalizada* o *softmax*, y corresponde a una generalización de la función logística a múltiples clases.
Además, esta función tiene la propiedad de ser una aproximación suave de la función máximo y convertir cualquier vector $s = [s_1, \ldots, s_k]$ en una distribución de probabilidad, donde podemos hablar de "la probabilidad de ser clase $\mathcal{C}_k$".

Regresión Logística

Regresión Logística

Analizaremos ahora los supuestos sobre el modelo generativo (i.e., las probabilidades de clase y condicionales) para encontrar un r que resulte en la bien conocida regresión logística.

Analizaremos ahora los supuestos sobre el modelo generativo (i.e., las probabilidades de clase y condicionales) para encontrar un r que resulte en la bien conocida regresión logística. Consideraremos el caso binario donde las densidades condicionales de clase son Gaussianas multivariadas, dadas por

$$p(x|\mathcal{C}_k) \sim \mathcal{N}(\mu_k, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)) \quad k \in \{1,2\}.$$

Analizaremos ahora los supuestos sobre el modelo generativo (i.e., las probabilidades de clase y condicionales) para encontrar un r que resulte en la bien conocida regresión logística. Consideraremos el caso binario donde las densidades condicionales de clase son Gaussianas multivariadas, dadas por

$$p(x|\mathcal{C}_k) \sim \mathcal{N}(\mu_k, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)) \quad k \in \{1,2\}.$$

Donde $u_k \in \mathbb{R}^M$ corresponde al centroide de la clase $\mathcal{C}_k$ y $\Sigma \in \mathbb{R}^{M\times M}$ simétrica y definida positiva, corresponde a la matriz de covarianza de las clases (misma matriz para todas las clases). Para este caso, se tiene que para la ecuación: $r = r(x) = \ln\left(\frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}\right)$

$$
\begin{aligned}
r &= \ln\left(\frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}\right) = \ln\left(\frac{\exp(-\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1))\mathbb{P}(\mathcal{C}_1)}{\exp(-\frac{1}{2}(x-\mu_2)^\top\Sigma^{-1}(x-\mu_2))\mathbb{P}(\mathcal{C}_2)}\right) \\
&= -\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^\top\Sigma^{-1}(x-\mu_2) + \ln\left(\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\right) \\
&= \frac{1}{2}\left(x^\top\Sigma^{-1}(\mu_1-\mu_2) + (\mu_1-\mu_2)^\top\Sigma^{-1}x - \mu_1\Sigma^{-1}\mu_1 + \mu_2\Sigma^{-1}\mu_2\right) + \ln\left(\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\right) \\
&= (\mu_1-\mu_2)^\top\Sigma^{-1}x + \frac{1}{2}\left(\mu_2\Sigma^{-1}\mu_2 - \mu_1\Sigma^{-1}\mu_1\right) + \ln\left(\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\right) \\
&= a^\top x + b
\end{aligned}
$$

$$
\begin{aligned}
r &= \ln\left(\frac{\mathbb{P}(x|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(x|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}\right) = \ln\left(\frac{\exp(-\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1))\mathbb{P}(\mathcal{C}_1)}{\exp(-\frac{1}{2}(x-\mu_2)^\top\Sigma^{-1}(x-\mu_2))\mathbb{P}(\mathcal{C}_2)}\right) \\
&= -\frac{1}{2}(x-\mu_1)^\top\Sigma^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^\top\Sigma^{-1}(x-\mu_2) + \ln\left(\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\right) \\
&= \frac{1}{2}\left(x^\top\Sigma^{-1}(\mu_1-\mu_2) + (\mu_1-\mu_2)^\top\Sigma^{-1}x - \mu_1\Sigma^{-1}\mu_1 + \mu_2\Sigma^{-1}\mu_2\right) + \ln\left(\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\right) \\
&= (\mu_1-\mu_2)^\top\Sigma^{-1}x + \frac{1}{2}\left(\mu_2\Sigma^{-1}\mu_2 - \mu_1\Sigma^{-1}\mu_1\right) + \ln\left(\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\right) \\
&= a^\top x + b
\end{aligned}
$$

donde hemos usado la notación

$$
\begin{aligned}
a &= \Sigma^{-1}(\mu_1-\mu_2) \\
b &= \frac{1}{2}(\mu_2^\top\Sigma^{-1}\mu_2 - \mu_1^\top\Sigma^{-1}\mu_1) + \ln\left(\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}\right).
\end{aligned}
$$

Regresión Logística

Lo anterior nos entrega la regresión logística (lineal) para el caso binario, donde al incorporar la expresión anterior en la ecuación logística obtenemos

$$p(\mathcal{C}_k|x) = \sigma(a^\top x + b) = \frac{1}{1 + \exp\left(-a^\top x - b\right)}.$$

Regresión Logística

Lo anterior nos entrega la regresión logística (lineal) para el caso binario, donde al incorporar la expresión anterior en la ecuación logística obtenemos

$$p(\mathcal{C}_k|x) = \sigma(a^\top x + b) = \frac{1}{1 + \exp\left(-a^\top x - b\right)}.$$

Ahora que hemos definido el modelo para nuestro problema de clasificación, aflora naturalmente la siguiente pregunta: ¿Cómo ajustar los parámetros de las condicionales a la clase y priors respectivamente? Para esto, reiteremos que los parámetros del modelos serán los de la probabilidad de clase $p(\mathcal{C}_k)$ y de la probabilidades condicionales de clase $p(x|\mathcal{C}_k)$. Respectivamente:

Regresión Logística

Lo anterior nos entrega la regresión logística (lineal) para el caso binario, donde al incorporar la expresión anterior en la ecuación logística obtenemos

$$p(\mathcal{C}_k|x) = \sigma(a^\top x + b) = \frac{1}{1 + \exp\left(-a^\top x - b\right)}.$$

Ahora que hemos definido el modelo para nuestro problema de clasificación, aflora naturalmente la siguiente pregunta: ¿Cómo ajustar los parámetros de las condicionales a la clase y priors respectivamente? Para esto, reiteremos que los parámetros del modelos serán los de la probabilidad de clase $p(\mathcal{C}_k)$ y de la probabilidades condicionales de clase $p(x|\mathcal{C}_k)$. Respectivamente:

- Probabilidad de clase:

$$p(\mathcal{C}_1) = \pi, \quad p(\mathcal{C}_2) = 1 - \pi,$$

  es decir, un parámetro π (por determinar).

Regresión Logística

Lo anterior nos entrega la regresión logística (lineal) para el caso binario, donde al incorporar la expresión anterior en la ecuación logística obtenemos

$$p(\mathcal{C}_k|x) = \sigma(a^\top x + b) = \frac{1}{1 + \exp\left(-a^\top x - b\right)}.$$

Ahora que hemos definido el modelo para nuestro problema de clasificación, aflora naturalmente la siguiente pregunta: ¿Cómo ajustar los parámetros de las condicionales a la clase y priors respectivamente? Para esto, reiteremos que los parámetros del modelos serán los de la probabilidad de clase $p(\mathcal{C}_k)$ y de la probabilidades condicionales de clase $p(x|\mathcal{C}_k)$. Respectivamente:

- Probabilidad de clase:

  $$p(\mathcal{C}_1) = \pi, \quad p(\mathcal{C}_2) = 1 - \pi,$$

  es decir, un parámetro π (por determinar).
- Probabilidad condicional de clase:

  $$p(x|\mathcal{C}_k) = \mathcal{N}(\mu_k, \Sigma); k = 1, 2$$

  es decir, parámetros $\mu_1 \in \mathbb{R}^M, \mu_2 \in \mathbb{R}^M, \Sigma \in \mathbb{R}^M \times \mathbb{R}^M$ (por determinar) o, equivalentemente, $M + M + M(M+1)/2 = M(M+5)/2$ parámetros escalares (considerando que Σ es simétrica).

Regresión Logística

Lo anterior nos entrega la regresión logística (lineal) para el caso binario, donde al incorporar la expresión anterior en la ecuación logística obtenemos

$$p(\mathcal{C}_k|x) = \sigma(a^\top x + b) = \frac{1}{1 + \exp\left(-a^\top x - b\right)}.$$

Ahora que hemos definido el modelo para nuestro problema de clasificación, aflora naturalmente la siguiente pregunta: ¿Cómo ajustar los parámetros de las condicionales a la clase y priors respectivamente? Para esto, reiteremos que los parámetros del modelos serán los de la probabilidad de clase $p(\mathcal{C}_k)$ y de la probabilidades condicionales de clase $p(x|\mathcal{C}_k)$. Respectivamente:

- Probabilidad de clase:

  $$p(\mathcal{C}_1) = \pi, \quad p(\mathcal{C}_2) = 1 - \pi,$$

  es decir, un parámetro π (por determinar).
- Probabilidad condicional de clase:

  $$p(x|\mathcal{C}_k) = \mathcal{N}(\mu_k, \Sigma); k = 1, 2$$

  es decir, parámetros $\mu_1 \in \mathbb{R}^M, \mu_2 \in \mathbb{R}^M, \Sigma \in \mathbb{R}^M \times \mathbb{R}^M$ (por determinar) o, equivalentemente, $M + M + M(M+1)/2 = M(M+5)/2$ parámetros escalares (considerando que Σ es simétrica).

Denotaremos todos los parámetros mediante el parámetro agregado $\theta = \{\pi, \mu_1, \mu_2, \Sigma\}$.

Regresión Logística

Realizaremos el entrenamiento del modelo mediante el método de máxima verosimilitud.

Regresión Logística

Realizaremos el entrenamiento del modelo mediante el método de máxima verosimilitud. Consideremos la codificación donde la observación (x_i, t_i) corresponde a clase $\mathcal{C}_1$ con $t_i = 1$ y a clase $\mathcal{C}_2$ con $t = 0$, podemos expresar la verosimilitud con una observación mediante:

$$L_i(\theta) = p(x_i, t_i|\theta) = p(x_i, \mathcal{C}_1|\theta)^{t_i} p(x_i, \mathcal{C}_0|\theta)^{1-t_i}.$$

Regresión Logística

Realizaremos el entrenamiento del modelo mediante el método de máxima verosimilitud. Consideremos la codificación donde la observación (x_i, t_i) corresponde a clase $\mathcal{C}_1$ con $t_i = 1$ y a clase $\mathcal{C}_2$ con $t = 0$, podemos expresar la verosimilitud con una observación mediante:

$$L_i(\theta) = p(x_i, t_i|\theta) = p(x_i, \mathcal{C}_1|\theta)^{t_i} p(x_i, \mathcal{C}_0|\theta)^{1-t_i}.$$

Para un conjunto de $\mathcal{D}$ de la forma

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{pmatrix} \in \mathbb{R}^{N \times M}, \qquad T = \begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix} \in \{0, 1\}^N \text{ es decir, codificación } 0 - 1.$$

Regresión Logística

Realizaremos el entrenamiento del modelo mediante el método de máxima verosimilitud. Consideremos la codificación donde la observación (x_i, t_i) corresponde a clase $\mathcal{C}_1$ con $t_i = 1$ y a clase $\mathcal{C}_2$ con $t = 0$, podemos expresar la verosimilitud con una observación mediante:

$$L_i(\theta) = p(x_i, t_i|\theta) = p(x_i, \mathcal{C}_1|\theta)^{t_i} p(x_i, \mathcal{C}_0|\theta)^{1-t_i}.$$

Para un conjunto de $\mathcal{D}$ de la forma

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_N^\top \end{pmatrix} \in \mathbb{R}^{N \times M}, \qquad T = \begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix} \in \{0,1\}^N \text{ es decir, codificación } 0-1.$$

podemos escribir la verosimilitud mediante $L(\theta) = p(X, T|\theta)$, luego:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{N} p(x_i, t_i|\theta) = \prod_{i=1}^{N} p(x_i, \mathcal{C}_1|\theta)^{t_i} p(x_i, \mathcal{C}_0|\theta)^{1-t_i} \\ &= \prod_{i=1}^{N} \left(p(x_i|\mathcal{C}_1, \theta)p(\mathcal{C}_1|\theta)\right)^{t_i} \left(p(x_i|\mathcal{C}_0, \theta)p(\mathcal{C}_0|\theta)\right)^{1-t_i} \\ &= \prod_{i=1}^{N} (\pi \mathcal{N}(x_i|\mu_1, \Sigma))^{t_i} ((1-\pi)\mathcal{N}(x_i|\mu_2, \Sigma))^{1-t_i}. \end{aligned}$$

Nuestro interés se encuentra en la log-verosimilitud

$$l(\theta) := \log L(\theta)$$

$$= \sum_{i=1}^{N} (t_i(\log(\pi) + \log(\mathcal{N}(x_i|\mu_1, \Sigma))) + (1 - t_i)(\log(1 - \pi) + \log(\mathcal{N}(x_i|\mu_2, \Sigma))))$$

Nuestro interés se encuentra en la log-verosimilitud

$$l(\theta) := \log L(\theta)$$

$$= \sum_{i=1}^{N} (t_i(\log(\pi) + \log(\mathcal{N}(x_i|\mu_1, \Sigma))) + (1 - t_i)(\log(1 - \pi) + \log(\mathcal{N}(x_i|\mu_2, \Sigma))))$$

Aplicando condiciones de primer orden

- **1)** Con respecto a π:

$$\frac{\partial \log(L)}{\partial \pi} = \sum_{i=1}^{N} \frac{t_i}{\pi} - \frac{1 - t_i}{1 - \pi} = 0$$

$$\Rightarrow \quad (1 - \pi) \sum_{i=1}^{N} t_i = \pi \sum_{i=1}^{N} (1 - t_i)$$

$$\Rightarrow \quad \sum_{i=1}^{N} t_i = \pi N \quad \Rightarrow \quad \pi = \frac{\sum_{i=1}^{N} t_i}{N} = \frac{N_1}{N_1 + N_2} \tag{6.1}$$

Donde $N_i := \text{Card}(x : x \in \mathcal{C}_i)$. Por lo tanto, el EMV de *pi* colapsa a la regla de Laplace.

- **2)** Con respecto a μ_1:

$$\frac{\partial \log(L)}{\partial \mu_1} = \sum_{i=1}^{N} t_i \frac{\partial}{\partial \mu_1}(-\frac{1}{2}(x_i - \mu_1)^\top \Sigma^{-1}(x_i - \mu_1))$$

$$= \sum_{i=1}^{N} t_i(\Sigma^{-1}(x_i - \mu_1)) = \Sigma^{-1}\sum_{i=1}^{N} t_i(x_i - \mu_1) = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} t_i x_i = \mu_1 \sum_{i=1}^{N} t_i \quad \Rightarrow \quad \mu_1 = \frac{1}{N_1}\sum_{i=1}^{N} t_i x_i = \frac{1}{N_1}\sum_{x_i \in C_1} x_i$$

De forma análoga:

$$\mu_2 = \frac{1}{N_2}\sum_{x_i \in C_2} x_i$$

- **2)** Con respecto a μ_1:

$$\frac{\partial \log(L)}{\partial \mu_1} = \sum_{i=1}^{N} t_i \frac{\partial}{\partial \mu_1}(-\frac{1}{2}(x_i - \mu_1)^\top \Sigma^{-1}(x_i - \mu_1))$$

$$= \sum_{i=1}^{N} t_i(\Sigma^{-1}(x_i - \mu_1)) = \Sigma^{-1}\sum_{i=1}^{N} t_i(x_i - \mu_1) = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} t_i x_i = \mu_1 \sum_{i=1}^{N} t_i \quad \Rightarrow \quad \mu_1 = \frac{1}{N_1}\sum_{i=1}^{N} t_i x_i = \frac{1}{N_1}\sum_{x_i \in C_1} x_i$$

De forma análoga:

$$\mu_2 = \frac{1}{N_2}\sum_{x_i \in C_2} x_i$$

Queda la siguiente pregunta, **¿Cuál es el estimador MV de Σ?**

Regresión Logística v/s modelo generativo

Regresión Logística v/s modelo generativo

Recordemos que los supuestos tomados sobre el modelo generativo para el problema de clasificación resultaron en:

$$p(\mathcal{C}_1|x) = \sigma(w^\top x) = \frac{1}{1 + e^{-w^\top x}}$$

donde por claridad de notación hemos elegido la representación lineal $(w^\top x)$ y no afín $(a^\top x + b)$.

Regresión Logística v/s modelo generativo

Recordemos que los supuestos tomados sobre el modelo generativo para el problema de clasificación resultaron en:

$$p(\mathcal{C}_1|x) = \sigma(w^\top x) = \frac{1}{1 + e^{-w^\top x}}$$

donde por claridad de notación hemos elegido la representación lineal $(w^\top x)$ y no afín $(a^\top x + b)$. En el caso anterior se ha entrenado el modelo generativo completo, es decir, $\pi, \mu_1, \mu_2, \Sigma$, lo cual tiene la ventaja de tener solución en forma cerrada, sin embargo, puede ser innecesario cuando solo necesitamos conocer el peso w en la ecuación anterior.

Regresión Logística v/s modelo generativo

Recordemos que los supuestos tomados sobre el modelo generativo para el problema de clasificación resultaron en:

$$p(\mathcal{C}_1|x) = \sigma(w^\top x) = \frac{1}{1 + e^{-w^\top x}}$$

donde por claridad de notación hemos elegido la representación lineal $(w^\top x)$ y no afín $(a^\top x + b)$. En el caso anterior se ha entrenado el modelo generativo completo, es decir, $\pi, \mu_1, \mu_2, \Sigma$, lo cual tiene la ventaja de tener solución en forma cerrada, sin embargo, puede ser innecesario cuando solo necesitamos conocer el peso w en la ecuación anterior. Calculemos la verosimilitud de la regresión logística con datos $\mathcal{D} = \{(x_i, t_i)\}_{i=1}^N$, para hacer la notación más compacta denotamos $\sigma_i = \sigma(w^\top x_i)$. Entonces:

$$p((t_i)_{i=1}^N|(x_i)_{i=1}^N, w) = \prod_{i=1}^N p(t_i|x_i, w) = \prod_{i=1}^N p(\mathcal{C}_1|x_i)^{t_i} p(\mathcal{C}_2|x_i)^{1-t_i} = \prod_{i=1}^N \sigma_i^{t_i}(1-\sigma_i)^{1-t_i}$$

Con lo que la log-verosimilitud está dada por

$$l(w) = \sum_{i=1}^{N} t_i \log(\sigma_i) + (1 - t_i) \log(1 - \sigma_i).$$

Con lo que la log-verosimilitud está dada por

$$l(w) = \sum_{i=1}^{N} t_i \log(\sigma_i) + (1 - t_i) \log(1 - \sigma_i).$$

Notemos que este problema de optimización no exhibe una solución en forma cerrada, por lo que podemos resolverlo mediante gradiente, para lo cual es necesario calcular el gradiente de $l(w)$ respecto a w:

$$\nabla_w l(w) = \sum_{i=1}^{N} (t_i - \sigma_i) x_i,$$

Con lo que la log-verosimilitud está dada por

$$l(w) = \sum_{i=1}^{N} t_i \log(\sigma_i) + (1 - t_i) \log(1 - \sigma_i).$$

Notemos que este problema de optimización no exhibe una solución en forma cerrada, por lo que podemos resolverlo mediante gradiente, para lo cual es necesario calcular el gradiente de $l(w)$ respecto a w:

$$\nabla_w l(w) = \sum_{i=1}^{N} (t_i - \sigma_i) x_i,$$

lo cual nos da una regla de ajuste $\theta \mapsto \theta - \eta \sum_{i=1}^{N} (\sigma_i - t_i) x_i$, o bien

$$\theta \mapsto \theta + \eta (t_i - \sigma_i) x_i,$$

si tomamos los datos de "a uno" (método del gradiente estocástico).

Regresión Logística v/s modelo generativo

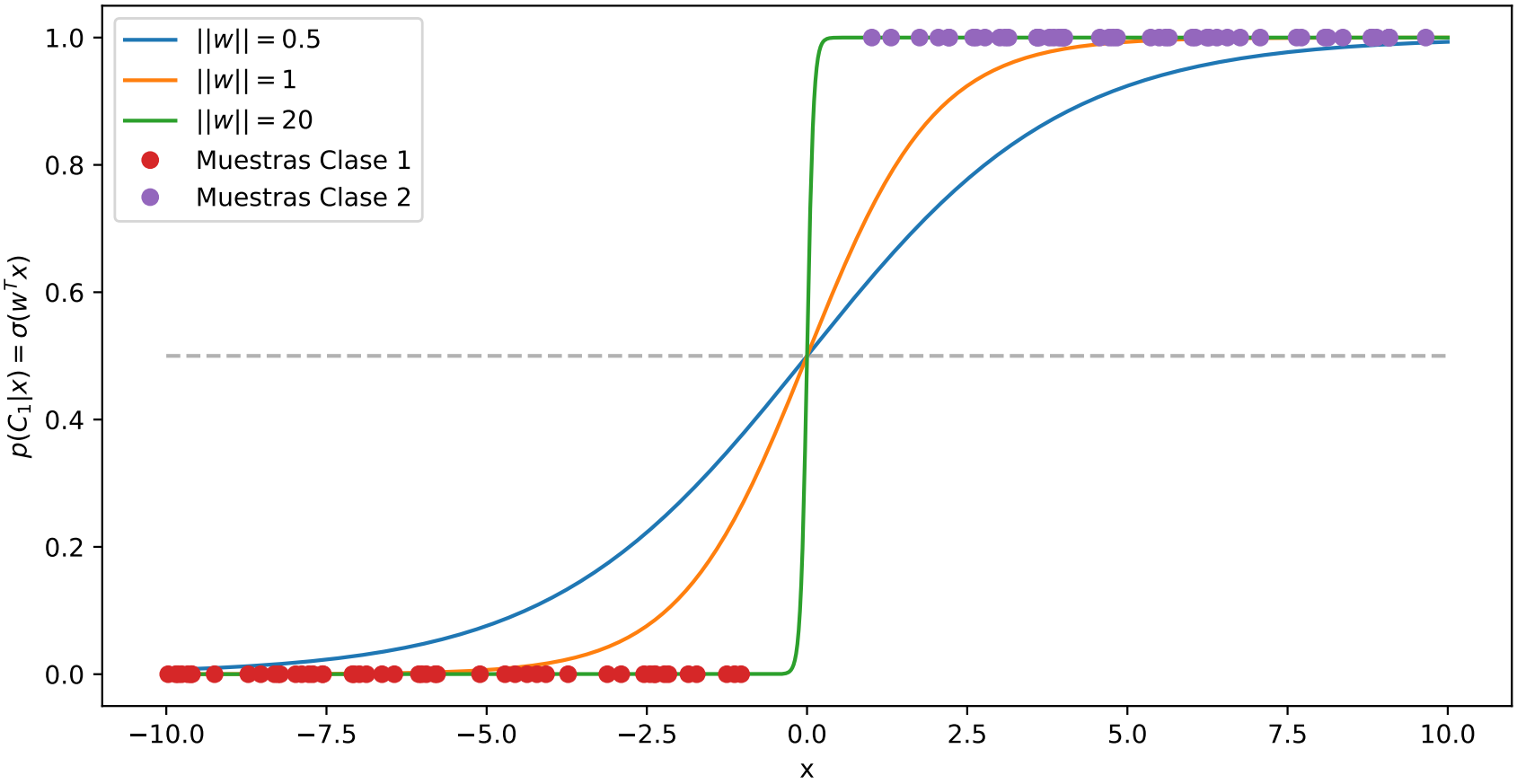

Fig. 4. En gris la frontera de decisión: una nueva entrada $x_\star$ será asignada a la clase $\mathcal{C}_1$ si $p(\mathcal{C}_1|x_\star) > \frac{1}{2}$, en caso contrario, será asignada a $\mathcal{C}_2$. Se observa que al entrenar con más muestras, $\|w\|$ crece por lo que el parámetro se sobreajusta a los datos y el clasificador converge a una función indicatriz.

Support Vector Machines

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.