Introducción al aprendizaje automático

•••

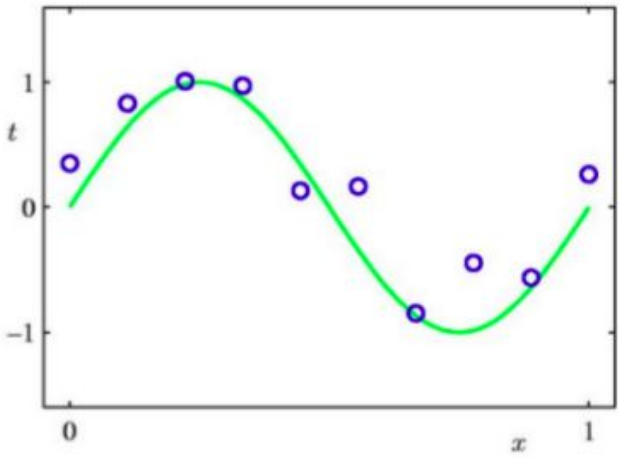
#2. Modelos probabilísticos y no paramétricos

Regresión

- Disponemos de N pares de entrenamiento (observaciones)

$$\{(x_i, y_i)\}_{i=1}^{N} = \{(x_1, y_1), \cdots, (x_N, y_N)\}$$

- El problema de regresión consiste en estimar $f(x)$ a partir de estos datos



Regresión polinomial

- **Función de predicción lineal:**

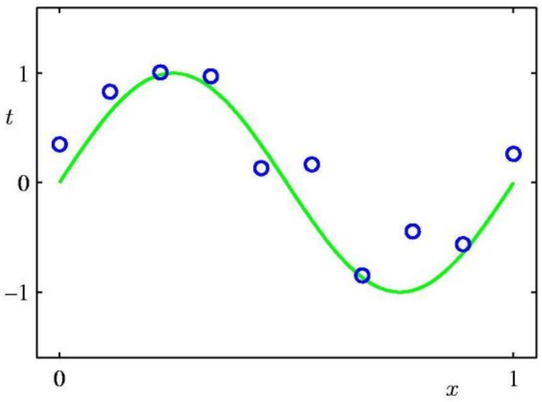
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M$$

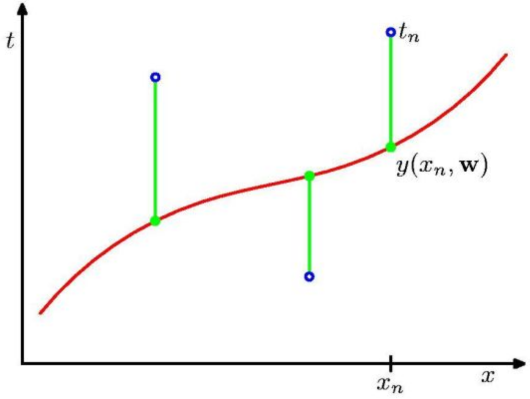
- **Función de costo**: error cuadrático

  medida del error en la predicción de t mediante $y(x; w)$

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

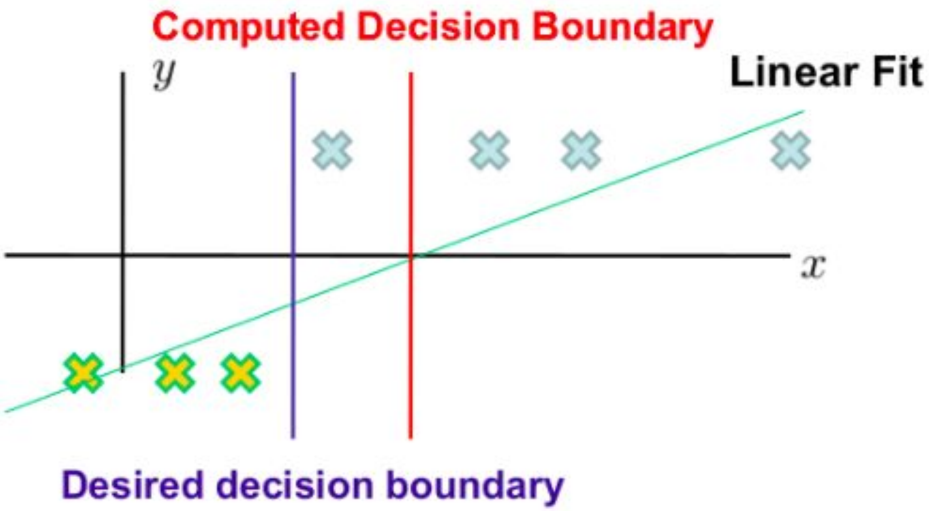
- ¿Se podría aplicar lo mismo en clasificación?





Error cuadrático en clasificación

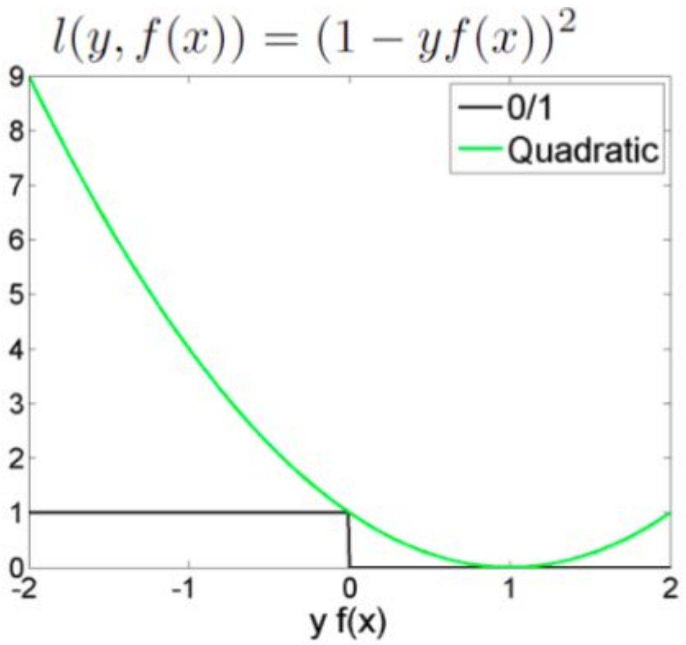
- Mínimo global único y solución en forma cerrada
- Pero, ¿es una medida del error de clasificación? ¿es adecuada?



Error cuadrático en clasificación

$$y_{\pm} \in \{-1, 1\}$$

$$\begin{aligned} l(y, f(x)) &= (y - f(x))^2 \\ &\overset{y^2=1}{=} y^2(y - f(x))^2 \\ &= (y^2 - yf(x))^2 \\ &\overset{y^2=1}{=} (1 - yf(x))^2 \end{aligned}$$

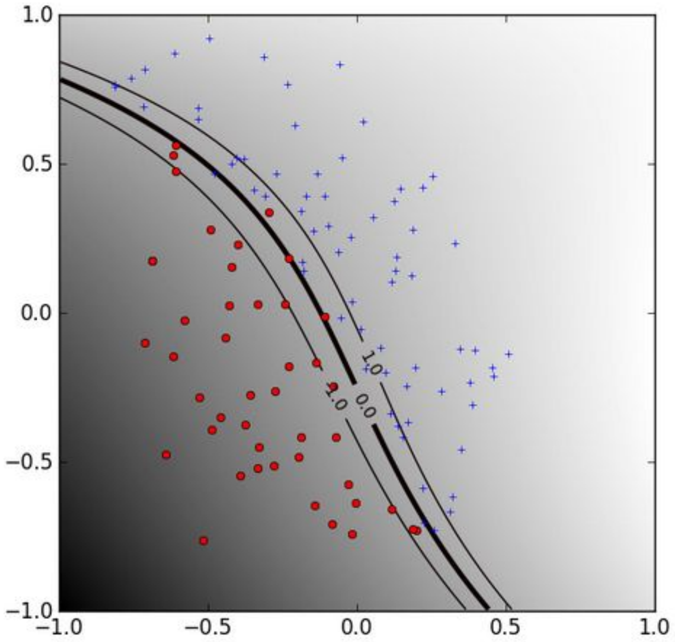


- No es robusta frente a *outliers*
- Penaliza predicciones que son muy buenas

Regresión logística

Clasificación basada en probabilidades

- Objetivo: dar una estimación de probabilidad de que una instancia x sea de una clase y, es decir, $p(y|x)$



- Recordar:

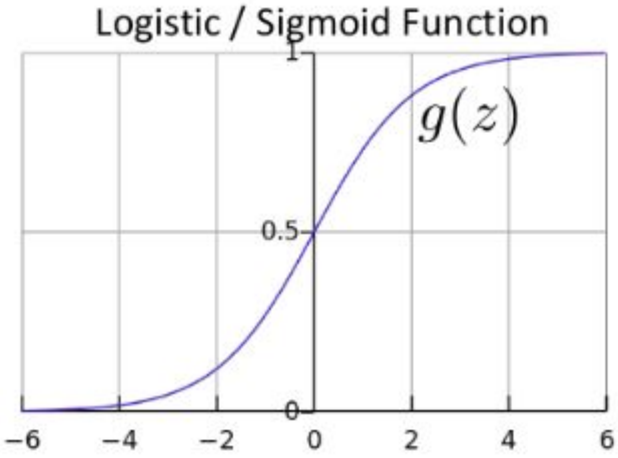
$$0 \leq p(evento) \leq 1$$

$$p(evento) + p(\neg evento) = 1$$

Regresión logística

- Aproximación probabilística al problema de clasificación
- La función de predicción $h_w(x)$ debe dar una aproximación de $p(y=1|x,w)$
- $0 \leq h_w(x) \leq 1$

$$h_w(x) = g(w^T x) = \frac{1}{1 + \exp(-w^T x)}$$



Regresión logística

- Dados $\left\{ \left(\boldsymbol{x}^{(1)}, y^{(1)} \right), \left(\boldsymbol{x}^{(2)}, y^{(2)} \right), \ldots, \left(\boldsymbol{x}^{(n)}, y^{(n)} \right) \right\}$

  donde $\boldsymbol{x}^{(i)} \in \mathbb{R}^d, \; y^{(i)} \in \{0, 1\}$

- Modelo: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}\right)$

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \qquad \boldsymbol{x}^\intercal = \begin{bmatrix} 1 & x_1 & \ldots & x_d \end{bmatrix}$$

Regresión logística. Función de costo

- ¿Y si usamos error cuadrático?

$$J(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2$$

pero el modelo de regresión logística no es lineal

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^{\top} \boldsymbol{x}}}$$

- El problema de optimización no tiene solución en forma cerrada

Regresión logística. Función de costo

- Conjunto de entrenamiento $\{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)\}, \quad \mathbf{x} \in R^M, \quad y \in \{0,1\}$

- y: observaciones discretas → muestras de una distribución Bernoulli

$$\begin{aligned} P(y=1|\mathbf{x},\mathbf{w}) &= f(\mathbf{x},\mathbf{w}) \\ P(y=0|\mathbf{x},\mathbf{w}) &= 1 - f(\mathbf{x},\mathbf{w}) \end{aligned}$$

$$P(y|\mathbf{x}) = (f(\mathbf{x},\mathbf{w}))^y \, (1 - f(\mathbf{x},\mathbf{w}))^{1-y}$$

- Encontrar el w que maximice la verosimilitud de las etiquetas en el conjunto de entrenamiento

$$\begin{aligned} -L(\mathbf{w}) = C(\mathbf{w}) &= \log P(\mathbf{y}|\mathbf{X},\mathbf{w}) = \sum_{i=1} \log P(y^i|\mathbf{x}^i,\mathbf{w}) \\ &= \sum_i y^i \log f(\mathbf{x}^i,\mathbf{w}) + (1-y^i)\log(1 - f(\mathbf{x}^i,\mathbf{w})) \end{aligned}$$

Función de costo. Intuición

La función de costo:

$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{n}\left[y^{(i)}\log h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) + \left(1-y^{(i)}\right)\log\left(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right)\right]$$

la podemos expresar como:

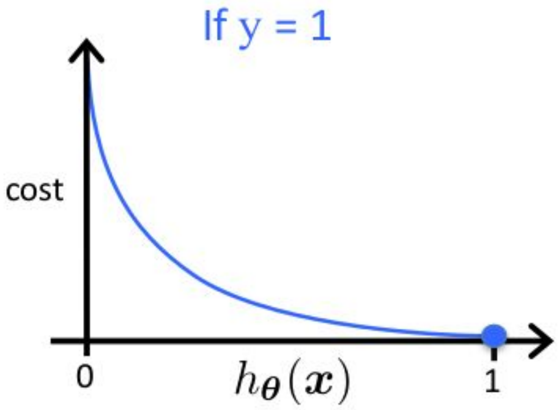
$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n}\mathrm{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}), y^{(i)}\right)$$

donde:

$$\mathrm{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 1 \\ -\log(1-h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 0 \end{cases}$$

Función de costo. Intuición

$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} \boxed{-\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 1} \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 0 \end{cases}$$

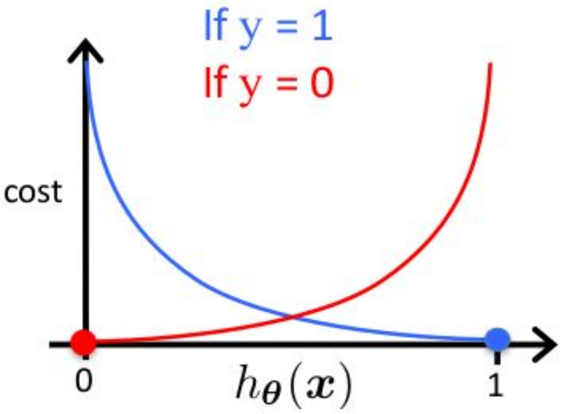


Caso y=1

- Costo 0 si la predicción es correcta
- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \rightarrow 0, \text{cost} \rightarrow \infty$
- Captura la intuición de que mayores errores deben recibir mayores penalizaciones

Función de costo. Intuición

$$\text{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) = \begin{cases} \boxed{-\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 1} \\ \boxed{-\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 0} \end{cases}$$



Caso y=0

- Costo 0 si la predicción es correcta
- $(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \rightarrow 0, \text{cost} \rightarrow \infty$
- Captura la intuición de que mayores errores deben recibir mayores penalizaciones

Regularización

$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{n}\left[y^{(i)}\log h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) + \left(1 - y^{(i)}\right)\log\left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right)\right]$$

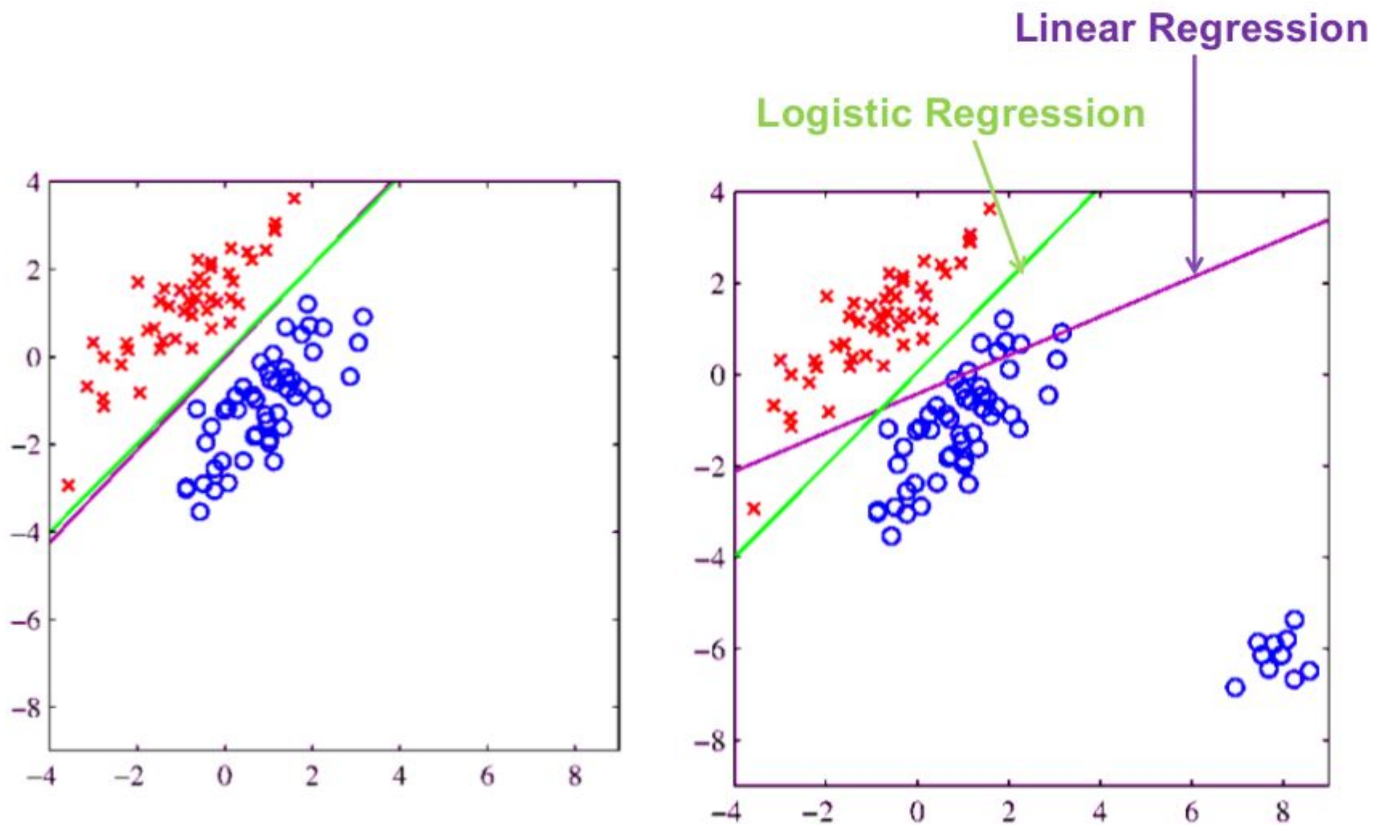
al igual que con regresión lineal:

$$\begin{aligned} J_{\text{regularized}}(\boldsymbol{\theta}) &= J(\boldsymbol{\theta}) + \lambda\sum_{j=1}^{d}\theta_j^2 \\ &= J(\boldsymbol{\theta}) + \lambda\|\boldsymbol{\theta}_{[1:d]}\|_2^2 \end{aligned}$$

$$\theta^* = \arg\min_{\theta} J(\theta)$$

[1:d] → excluir el término constante

Regresión lineal vs. regresión logística



Naïve Bayes

Regla de Bayes

- Dos formas de factorizar una distribución en dos variables:

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$

- Operando:

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

- ¿Porqué es útil?
  - Nos permite "revertir" el condicional
  - A veces una dirección es difícil de calcular, pero la otra no
  - Es la base de muchos modelos

El clasificador de Bayes

- Distribución conjunta sobre $X_1, \ldots, X_n$ e Y

- Podemos definir una función de predicción de la forma:

$$\arg\max_Y P(Y|X_1, \ldots, X_n)$$

  - por ejemplo: ¿cuál es la probabilidad de que una imagen represente un "5" dado el valor de sus píxeles?

- Problema: ¿cómo computamos $\mathrm{P}(Y|X_1, \ldots, X_n)$? …

El clasificador de Bayes

- … ¡Usando regla de Bayes!

$$P(Y|X_1,\ldots,X_n) = \frac{\overbrace{P(X_1,\ldots,X_n|Y)}^{\text{Likelihood}}\overbrace{P(Y)}^{\text{Prior}}}{\underbrace{P(X_1,\ldots,X_n)}_{\text{Normalization Constant}}}$$

  - Ahora podemos pensar en modelar cómo los píxeles de la imágen son "generados" dado el número "5".

Naïve Bayes

- Hipótesis: los X_i son independientes dado Y

$$
\begin{aligned}
P(X_1, X_2|Y) &= P(X_1|X_2, Y)P(X_2|Y) \\
&= P(X_1|Y)P(X_2|Y)
\end{aligned}
$$

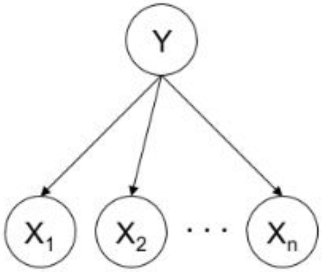
- O en forma más general:

$$\boxed{P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)}$$

- Si los X_i consisten en n valores binarios, ¿cuántos parámetros necesito especificar para $P(X_i|Y)$?

El clasificador naïve Bayes

- Dado:
  - Distribución a priori $P(Y)$
  - n features X_i condicionalmente independientes dada la clase Y



- Para cada X_i, especificar $P(X_i \mid Y)$

- Función de decisión:

$$
\begin{aligned}
y^* = h_{NB}(\mathbf{x}) &= \arg\max_y P(y) P(x_1, \ldots, x_n \mid y) \\
&= \arg\max_y P(y) \prod_i P(x_i \mid y)
\end{aligned}
$$

Estimación de parámetros por MV

- Dado un conjunto de datos, obtener Count(A=a, B=b) , es decir, el número de ejemplos en donde A=a y B=b.
- MV para naïve Bayes sobre variables discretas:
  - Prior:

$$P(Y=y) = \frac{Count(Y=y)}{\sum_{y'} Count(Y=y')}$$

  - Distribución condicionales (observación):

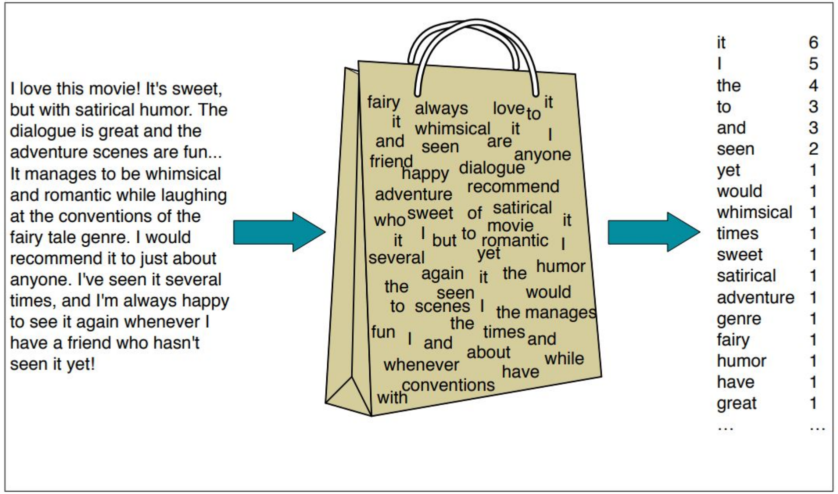
$$P(X_i=x|Y=y) = \frac{Count(X_i=x, Y=y)}{\sum_{x'} Count(X_i=x', Y=y)}$$

Ejemplo: Clasificación de texto

- Clasificación temática de artículos
  - ¿Habla de política o de deportes? ¿Es un paper de microbiología o de física cuántica?
- Atribución de autoría / detección de plagio
  - ¿Quién escribió esto? ¿Es quien dice ser?
- Análisis de Sentimiento
  - ¿Habla a favor o en contra? ¿Le gustó o no le gustó?
- Detección de discurso de odio/discriminatorio/toxicidad
  - ¿Es discriminatorio? ¿A qué grupo discrimina? ¿Llama a la acción?
- Identificación de idioma / región
  - ¿Es castellano o portugués? ¿Es jujeño o cordobés?

Representación con Bolsas de Palabras

- Forma tradicional en PLN de codificar texto en vectores (pre word embeddings)
- Cada palabra es un feature: el valor indica la cantidad de veces que aparece
- Alta dimensionalidad: vectores del tamaño del vocabulario
- Dispersos: muchísimos ceros



Bolsas de Palabras: Ejemplo

Índice		“el mejor guión”	“no es buena”	“de lo mejor”
0	de	0	0	1
1	es	0	1	0
2	no	0	1	0
3	buena	0	1	0
4	mejor	1	0	1
5	patética	0	0	0
6	guión	1	0	0

Regla de Bayes con Documentos y Clases

- Para un documento d y una clase c:

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

El Clasificador Naive Bayes

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP: "Maximum a posteriori"
= clase más probable

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c) P(c)}{P(d)}$$

Regla de Bayes

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

¡El denominador
no depende de c!
Lo descarto

El Clasificador Naive Bayes (cont.)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} \, P(d \mid c) P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} \, P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Documento d representado como features $x_1, \ldots, x_n$

El Clasificador Naive Bayes (cont.)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} \underbrace{P(x_1, x_2, \ldots, x_n \mid c)}_{|X|^n \,*\, C \text{ parámetros!!}} \underbrace{P(c)}_{\text{¿qué tan frecuente es la clase?}}$$

|X|^n * C parámetros!!

sólo se podría estimar con un número muy muy grande de ejemplos de entrenamiento

¿qué tan frecuente es la clase?

simplemente frecuencia relativa

El Clasificador Naive Bayes (cont.)

Suposiciones de independencia

- **Bag of Words:** el orden de las palabras no importa.
- **Independencia condicional “inocente” (naive):**

  Las probabilidades de los features son independientes entre sí:

$$P(x_1,\ldots,x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

El Clasificador Naive Bayes (cont.)

Entonces queda:

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \underbrace{P(c)}_{\text{Prior}} \underbrace{\prod_{x \in X} P(x \mid c)}_{\text{Likelihood}}$$

Aprendizaje: Máxima Verosimilitud

- Simplemente calcular frecuencias relativas:

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

Prior: frecuencia relativa de cada clase.

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Likelihood: frecuencia relativa de la palabra w_i en todos los documentos de clase c_j.

Problema con Máxima Verosimilitud

- ¿Qué pasa si nunca vimos en entrenamiento una palabra en particular en los documentos de una clase? Ejemplo:

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{count(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} count(w, \text{positive})} = 0$$

- Alcanza con un término cero para que todo sea cero:

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c_j) \prod_{x \in X} P(x \mid c)$$

“Add-1”: Suavizado de Laplace para Naive Bayes

- Hacemos de cuenta que vimos al menos una vez todas las palabras:

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c) + 1 \right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c) \right) + |V|}$$

- ¡Ya no hay más ceros!

Ejemplo Detallado: ¿Habla de china o de japón?

Dan Jurafsky

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Choosing a class:

$P(c|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14$
≈ 0.0003

Conditional Probabilities:

$P(\text{Chinese}|c) = (5+1) / (8+6) = 6/14 = 3/7$

$P(\text{Tokyo}|c) = (0+1) / (8+6) = 1/14$

$P(\text{Japan}|c) = (0+1) / (8+6) = 1/14$

$P(\text{Chinese}|j) = (1+1) / (3+6) = 2/9$

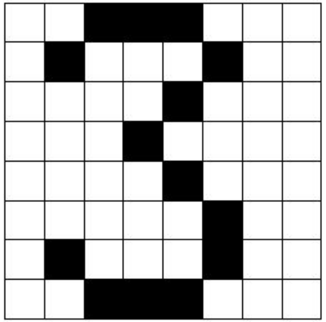
$P(\text{Tokyo}|j) = (1+1) / (3+6) = 2/9$

$P(\text{Japan}|j) = (1+1) / (3+6) = 2/9$

$P(j|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9$
≈ 0.0001

Otro ejemplo: reconocimiento de dígitos

- Input: pixel grids



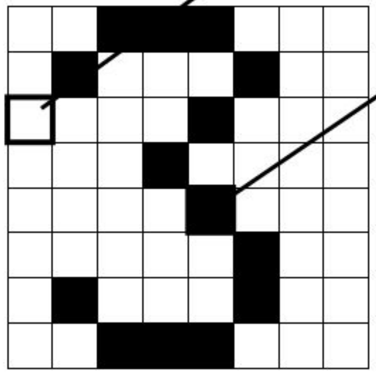
- Output: a digit 0-9

Pregunta: ¿cuán realista es la hipótesis del clasificador naïve Bayes en este ejemplo?

Otro ejemplo: reconocimiento de dígitos

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

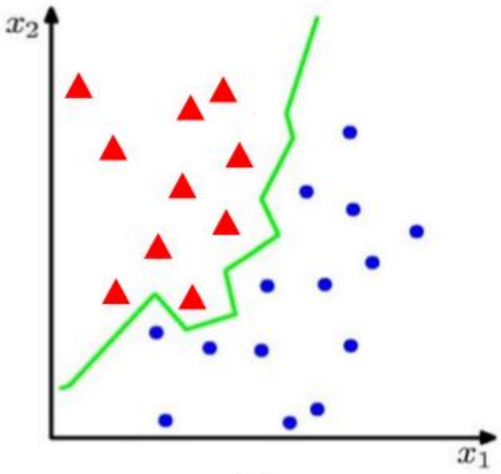
$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

Modelos no paramétricos: vecinos más cercanos

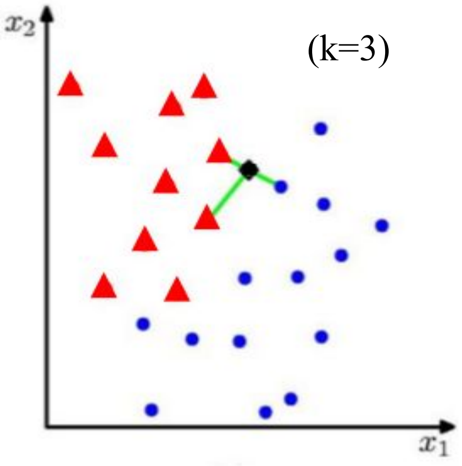
Clasificación (binaria)

- Dado un conjunto de datos de entrenamiento $D=\{(\mathbf{x}_i, y_i), i=1, ..., N\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$.
- Encontrar una función a partir de D tal que $f(\mathbf{x}_i) \approx y_i$



Clasificador k vecinos más próximos (*k*-NN)

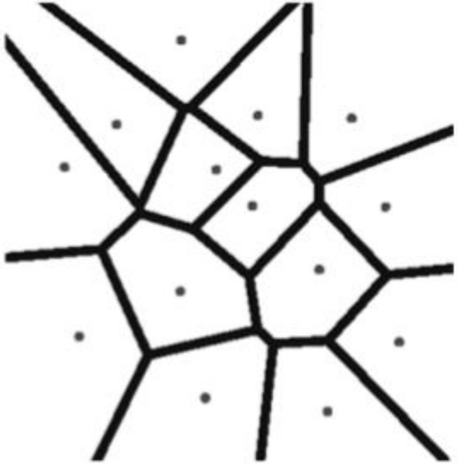
- Dado un conjunto de datos de entrenamiento $\mathrm{D}=\{(\mathbf{x}_i, y_i), i=1, ..., N\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$.

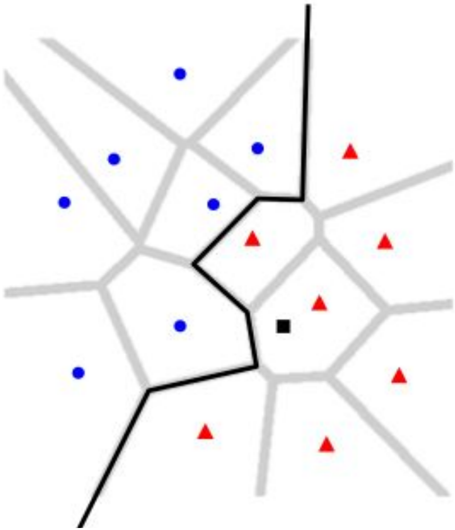


Algoritmo:

- Dado un punto de test $\mathbf{x}$, encontrar sus *k* vecinos más próximos en D.
- Asignar la clase mayoritaria en el conjunto de vecinos

Caso especial *k*=1





Diagramas de Voronoi

- Partición del espacio en regiones disjuntas.
- Frontera entre regiones definidas por puntos equidistantes a puntos de entrenamiento.

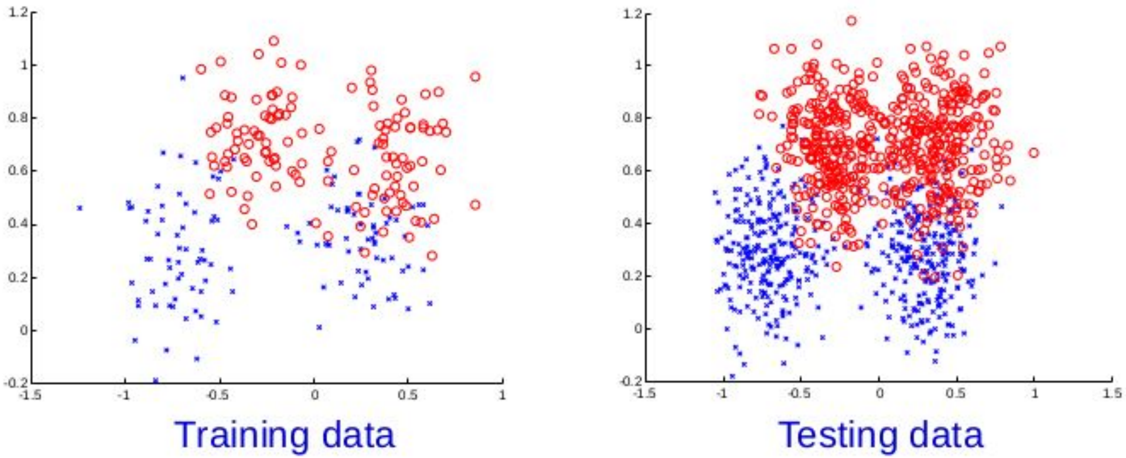
Clasificación

- Frontera de decisión no lineal
- Extensión a multiclase trivial (con algunas heurísticas)

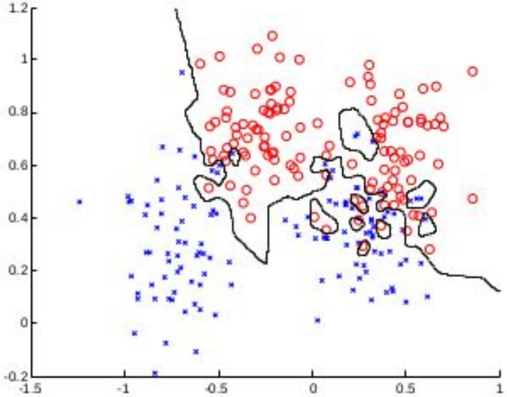
Análisis

- Asumimos que los conjuntos de entrenamiento y test son muestras independientes de la misma distribución (universal)

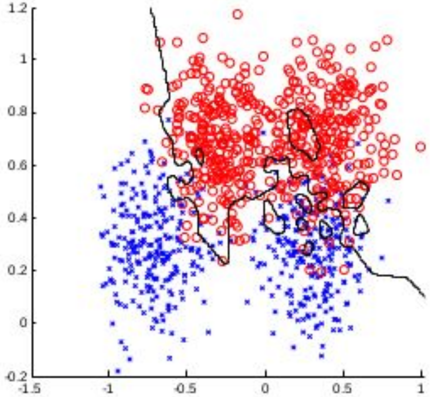
- Medida del error de clasificación: $\frac{1}{N}\sum_{i=1}^{N}[y_i \neq f(\mathbf{x}_i)]$



k=1

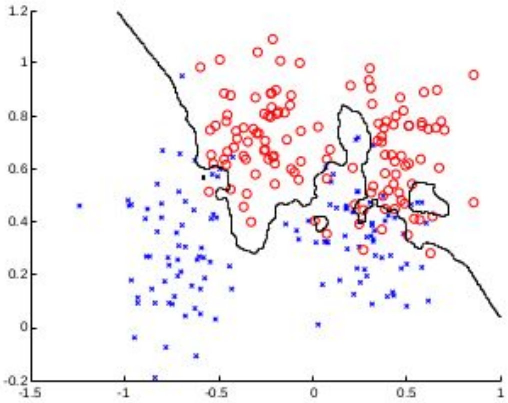


error = 0.0

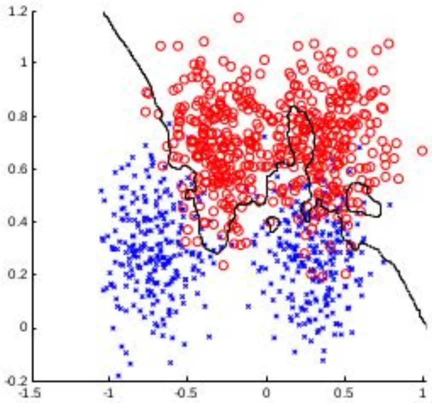


error = 0.15

k=3

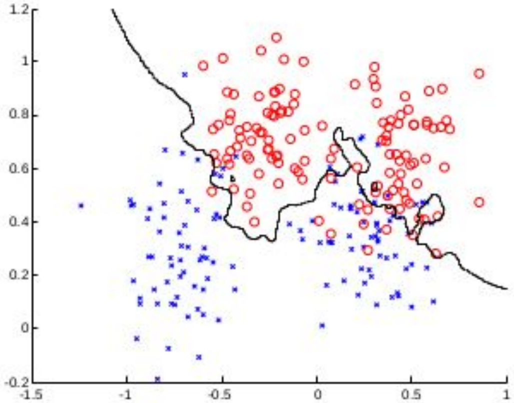


error = 0.0760

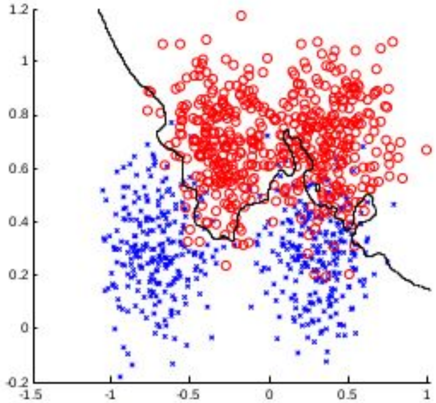


error = 0.1340

k=7

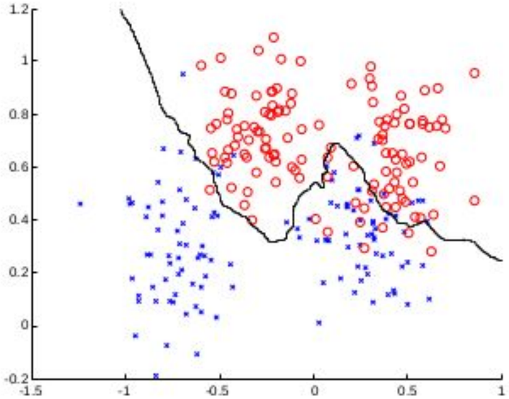


error = 0.1320

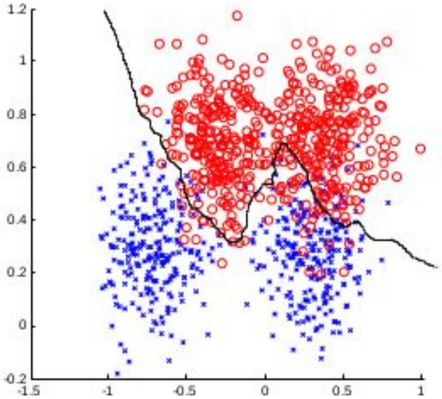


error = 0.1110

k=21



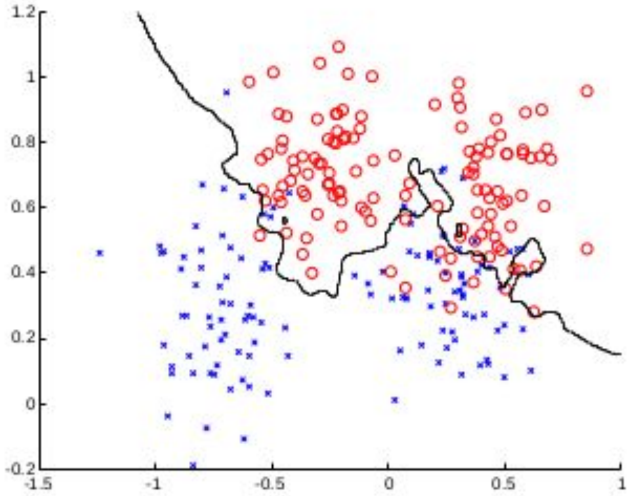
error = 0.1120



error = 0.0920

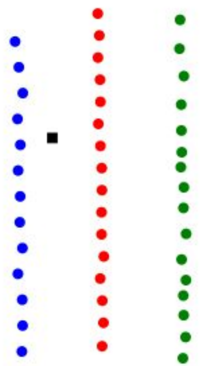
Ventajas

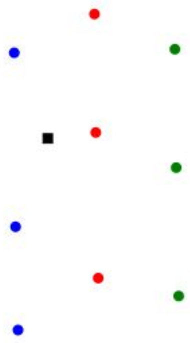
- *k*-NN es un método simple y efectivo
- aplicable a problemas multiclase
- Frontera de decisión no lineal
- La calidad de las predicciones mejora con más datos de entrenamiento
- Solo un hiperparámetro, K



Desventajas

- Necesidad de definir una métrica/distancia
- Costo computacional
  - Se deben almacenar los ejemplos de entrenamiento
  - Cada muestra se debe comparar con todas las de entrenamiento
- Búsqueda aproximada, estructuras de datos, *thining*, ...





Problemas multiclase

Clasificación multiclase

- Una muestra puede pertenecer a 1 (o más) de K clases
    - Datos de entrenamiento $\{(\mathbf{x}_i, y_i)\}$, y_i=1,…, K

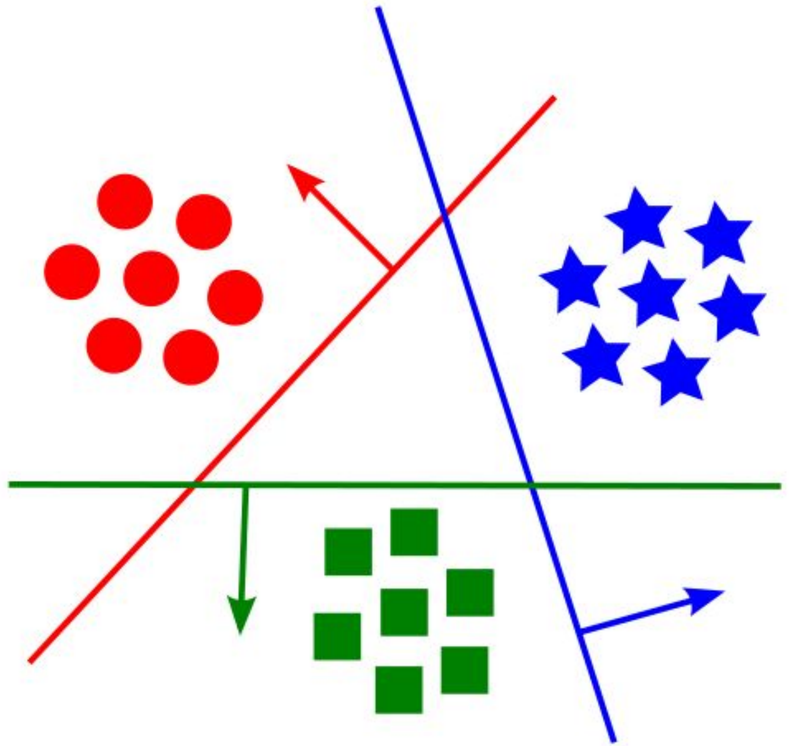
- Distintos tipos de problemas:
    - multiclase: $\mathbf{x}$ pertenece solo a una categoría
    - multietiqueta: $\mathbf{x}$ puede pertenecer a más de una categoría

- A veces es más fácil descomponer el problema multiclase en una serie de problemas binarios. **Distintas estrategias: OVA, AVA, ...**

Estrategia uno contra todos (OVA)

- **Asumimos que cada clase es separable del resto**
- Dado un conjunto de entrenamiento $D=\{(\mathbf{x}_i, y_i)\}$, y_i =1,...,K
  - Descomponer el problema en K problemas binarios. Para la clase *k*, crear un problema tal que:
    - Ejemplos cuya etiqueta es y_i =k son ejemplos positivos
    - Ejemplos cuya etiqueta es $y_i \neq$k son ejemplos negativos
  - Generar K clasificadores binarios con **función de predicción** $f_k(\mathbf{x})$, k=1,...,K.
- Predicción (*winner takes all*): k* = argmax $f_k(\mathbf{x})$

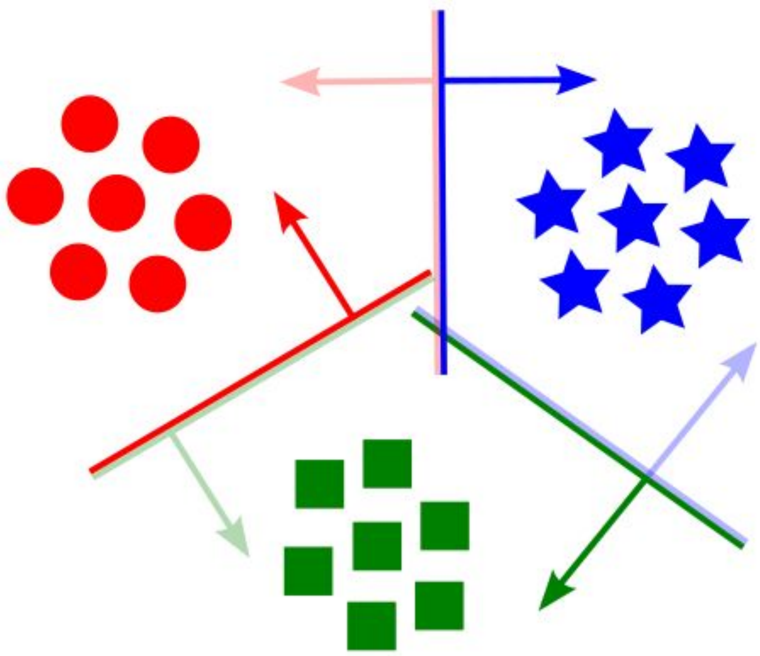
Estrategia uno contra todos (OVA)



Estrategia todos contra todos (AVA)

- **Asumimos que cada clase par de clases es separable**
- Dado un conjunto de entrenamiento $D=\{(\mathbf{x}_i, y_i)\}$, y_i =1,...,K
    - Descomponer el problema en K(K-1)/2 problemas binarios. Para el par de clases (i, j), $i \neq j$, crear un problema tal que:
        - Ejemplos cuya etiqueta es $y_i = i$ son ejemplos positivos
        - Ejemplos cuya etiqueta es $y_i = j$ son ejemplos negativos
    - Generar K(K-1)/2 clasificadores binarios con **función de decisión** $g_{(i,\, j)}(\mathbf{x})$
- Predicción (*voting*): cada clase recibe K-1 "votos"

Estrategia todos contra todos (AVA)



Regresión logística multiclase

- Para dos clases:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^{\top}\boldsymbol{x})} = \frac{\exp(\boldsymbol{\theta}^{\top}\boldsymbol{x})}{\boxed{1} + \boxed{\exp(\boldsymbol{\theta}^{\top}\boldsymbol{x})}}$$

peso asignado
a y=0

peso asignado
a y=1

- Para C clases (c=1,...,C):

$$p(y = c \mid \boldsymbol{x}; \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C) = \frac{\exp(\boldsymbol{\theta}_c^{\top}\boldsymbol{x})}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^{\top}\boldsymbol{x})}$$

(función **softmax**)