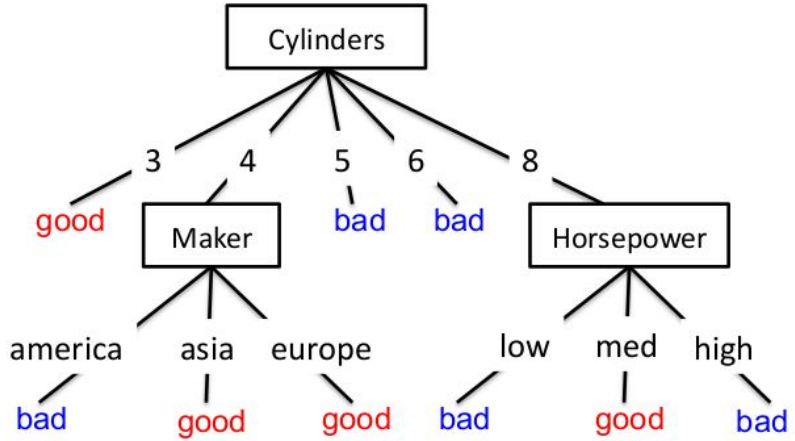
Módulo 3. IA y grandes volúmenes de datos

#4. Árboles de decisión y modelos probabilísticos

Árboles de decisión en clasificación

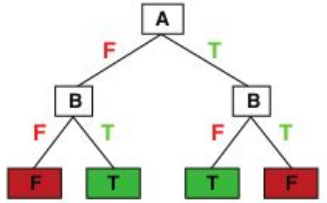
- Cada nodo interno compara un atributo x_i
- Una rama por cada valor de atributo $x_i = v$
- Cada hoja asigna una clase y
- Para clasificar un $\boldsymbol{x}$, atravesar el árbol de tronco a hojas y devolver el y asignado
- Modelo interpretable



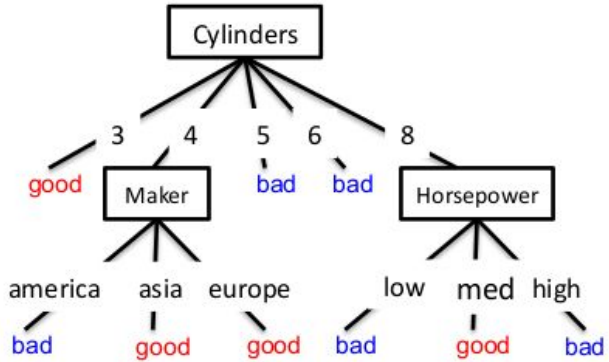
¿Qué funciones se pueden representar?

- Pueden representar cualquier función de los atributos de entrada
- Para funciones booleanas, un camino de tronco a hoja define una fila en la tabla de verdad
- Puede requerir un número exponencial de nodos

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



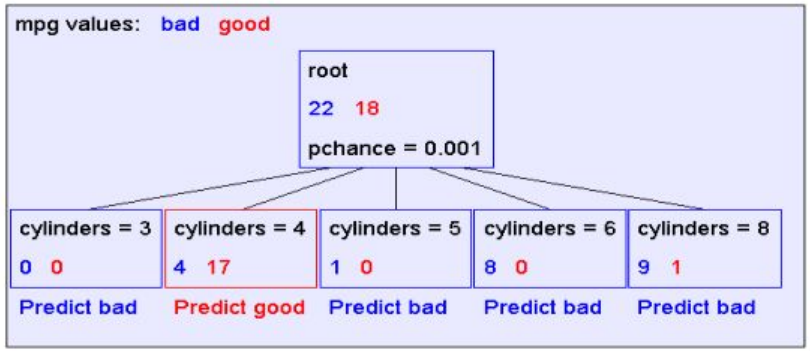
(Figure from Stuart Russell)



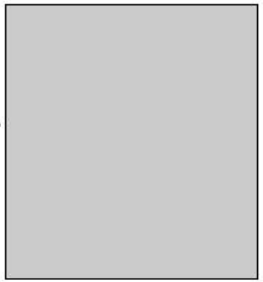
Complejidad y aprendizaje

- Aprender el árbol de decisión más simple (más chico) es un problema NP-completo (Hyafil & Rivest, 1976)
- Debemos recurrir a heurísticas voraces (*greedy*)
  - Comenzar con un árbol vacío
  - Generar una partición usando **siguiente mejor atributo**
  - Paso anterior de forma recursiva

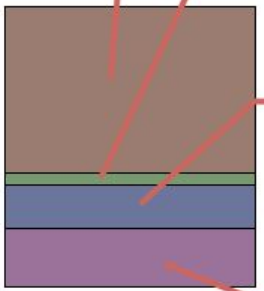
Aprendizaje de árboles de forma recursiva



Take the Original Dataset..



And partition it according to the value of the attribute we split on



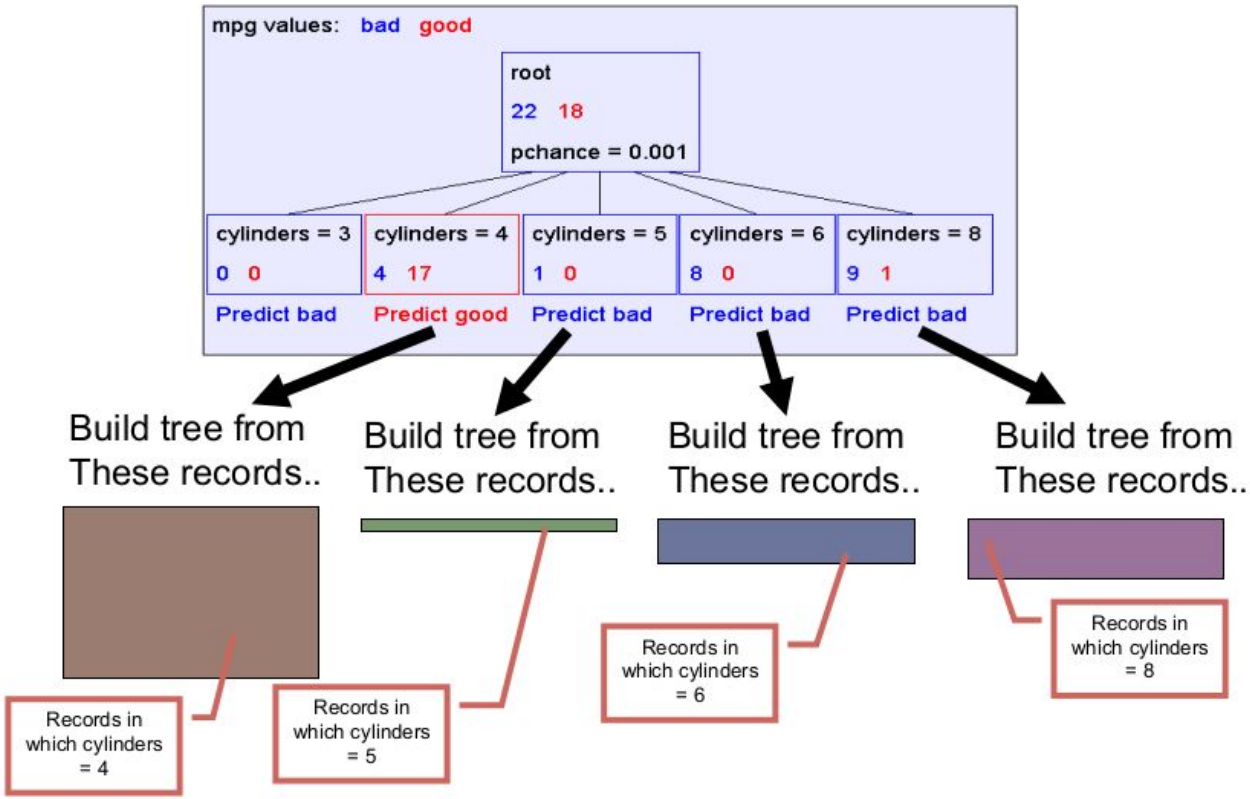
Records in which cylinders = 4

Records in which cylinders = 5

Records in which cylinders = 6

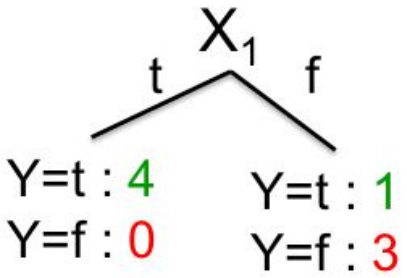
Records in which cylinders = 8

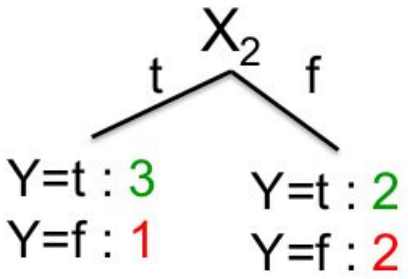
Paso recursivo



Particionado: elegir un buen atributo

Preferiríamos partir usando X_1 o X_2?





X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Medida de incertidumbre

- Una partición es buena si estamos “más seguros” de la clasificación después de haberla realizado
  - Determinística (indicatriz) = bueno
  - Uniforme = malo
  - ¿Qué pasa con distribuciones intermedias?

P(Y=A) = 1/2	P(Y=B) = 1/4	P(Y=C) = 1/8	P(Y=D) = 1/8

P(Y=A) = 1/4	P(Y=B) = 1/4	P(Y=C) = 1/4	P(Y=D) = 1/4

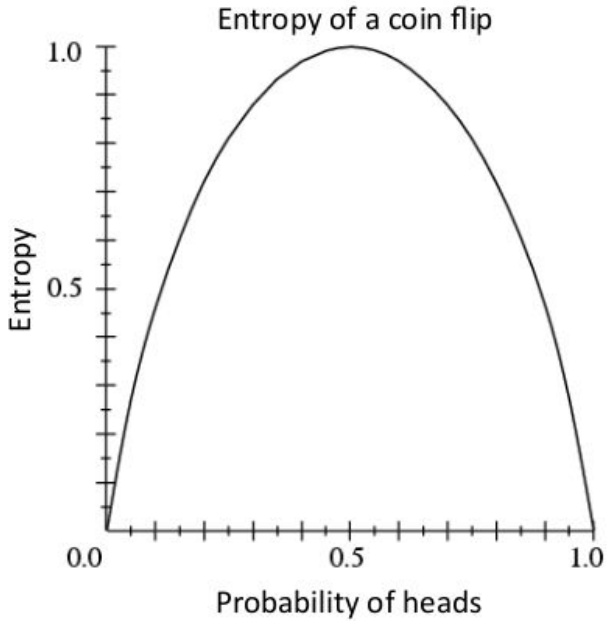
Entropía

- La entropía $\mathrm{H(Y)}$ de una variable aleatoria discreta Y

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

- **A mayor incertidumbre, mayor entropía.**

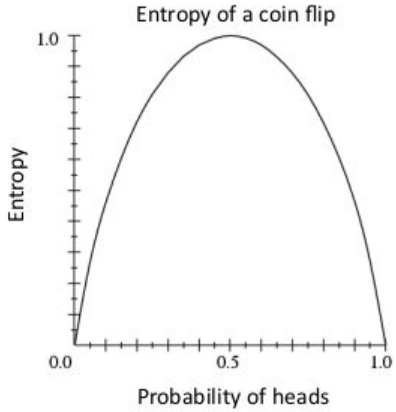
- Interpretación según *teoría de la información*:
  H(Y) es el número esperado de bits necesarios para codificar un valor aleatorio de Y



Entropía

- Entropía alta
  - Y proviene de una distribución más uniforme
  - Histograma chato
  - Muestras de Y son menos predecibles
- Entropía baja
  - Y proviene de una distribución más variada (picos y valles)
  - Histogramas más irregulares
  - Muestras de Y son más predecibles

Ejemplo



$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

P(Y=t) = 5/6

P(Y=f) = 1/6

H(Y) = - 5/6 $\log_2$ 5/6 - 1/6 $\log_2$ 1/6

= 0.65

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

Entropía condicional

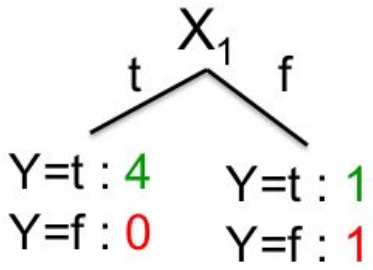
- Entropía condicional H(Y|X) de una v.a. Y condicionada a una v.a. X

$$H(Y \mid X) = -\sum_{j=1}^{v} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i \mid X = x_j) \log_2 P(Y = y_i \mid X = x_j)$$

Example:

$P(X_1=t) = 4/6$

$P(X_1=f) = 2/6$



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$H(Y|X_1) = -\ 4/6\ (1 \log_2 1 + 0 \log_2 0)$

$- 2/6\ (1/2 \log_2 1/2 + 1/2 \log_2 1/2)$

$= 2/6$

Ganancia de información

- Decrecimiento de entropía (incertidumbre) luego de la partición

$$IG(X) = H(Y) - H(Y \mid X)$$

Del ejemplo:

$IG(X_1) = H(Y) - H(Y|X_1)$
$= 0.65 - 0.33$

$IG(X_1) > 0 \rightarrow$ Elegimos X_1

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

Aprendizaje de árboles de decisión

- Comenzar con un árbol vacío
- Generar una partición usando **siguiente mejor atributo**
  - Usar, por ejemplo, ganancia de información:

$$\arg\max_i IG(X_i) = \arg\max_i H(Y) - H(Y \mid X_i)$$

  - Paso anterior de forma recursiva

Sobreajuste en árboles de decisión

- El error de entrenamiento es siempre cero (si no hay errores en las etiquetas)
- Poca capacidad de generalización
- Se debe inducir algún sesgo a modelos más simples
  - Fijar un límite a la profundidad del árbol
  - Número mínimo de muestras en cada nodo hoja
  - etc …
- Ensamble: Random Forests (bosques!) y boosted trees

Entradas con valores reales

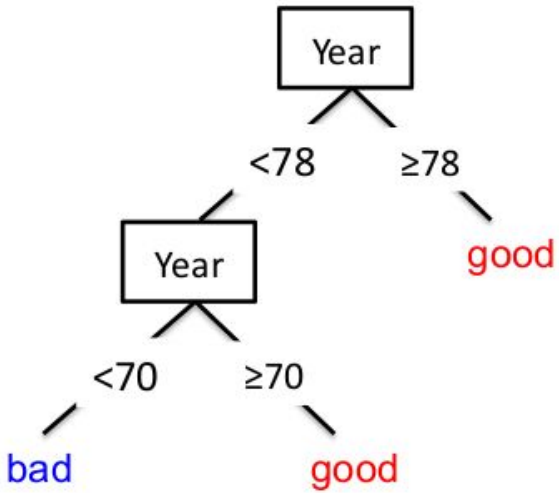
- Existe un número infinito de posibles particiones!

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europe
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europe
bad	5	131	103	2830	15.9	78	europe

Partición mediante umbrales

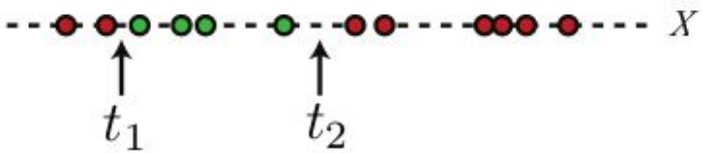
- Árboles binarios
  - partir un atributo X a un valor *t*
    - Una rama para x<*t*
    - Una rama para x≥*t*

- Se deben permitir particiones de un mismo atributo en distintos niveles de un mismo camino



El conjunto de posibles umbrales

- Árboles binarios, atributo X
  - Una rama para x<*t*
  - Una rama para x≥*t*
- Explorar todos los valores posibles de *t* es intratable
- Solo un número finito de valores es importante
  - Ordenar X de acuerdo a los valores del atributo $\{x_1, \ldots, x_N\}$
  - Considerar puntos (umbrales) de la forma $x_i+(x_{i+1}-x_i)/2$
  - Considerar puntos (umbrales) entre muestras de clases distintas



Elegir el mejor umbral

- Supongamos una variable X y umbral *t*
- IG(Y|X:t) denota la ganancia de información para Y cuando particionamos X de acuerdo a *t*
- Definimos:

$$H(Y|X:t) = p(X < t)\, H(Y|X < t) + p(X >= t)\, H(Y|X >= t)$$
$$IG(Y|X:t) = H(Y) - H(Y|X:t)$$
$$IG^*(Y|X) = \max_t IG(Y|X:t)$$

- Usamos $IG^*(Y|X)$ con variables continuas

Árboles de decisión. Resumen

- Uno de los modelos más utilizados en la práctica
  - Fáciles de comprender, implementar y utilizar
  - Computacionalmente eficientes
- Muchas variantes para selección de atributos basados en ganancia de información (ID3, C4.5, …)
- Se pueden utilizar en regresión y para la estimación de densidades
- Sobreajuste por definición!
  - Heurísticas para definir árboles más simples (*pruning*, *fixed depth*, *early stopping*, etc)
  - *ensemble* de distintos árboles (eg. *random forests*)

Modelos probabilísticos: 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:

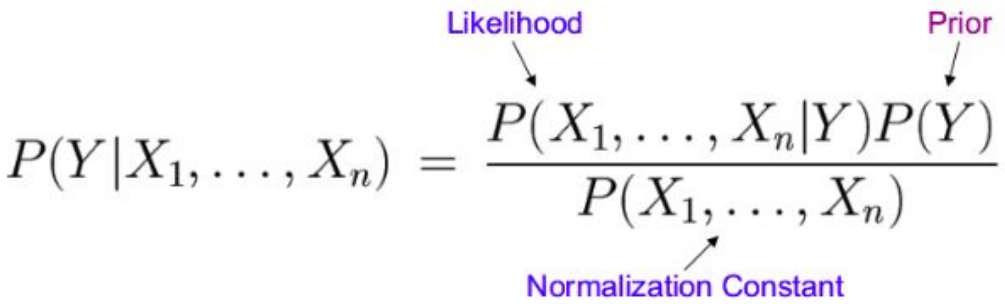
$$\arg\max_Y P(Y|X_1, \ldots, X_n)$$

  - 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!



  - 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

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

- O en forma más general:

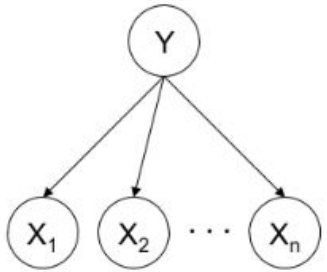
$$P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$$

- 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 $\mathrm{P}(Y)$
  - n features X_{i} condicionalmente independientes dada la clase Y

- Para cada X_{i}, especificar $\mathrm{P}(X_{\mathrm{i}} \mid Y)$



- 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:

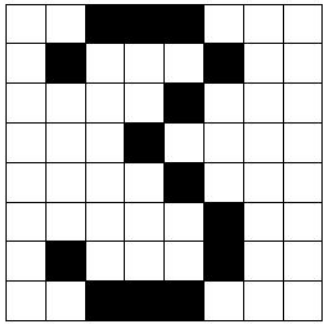
$$P(Y = y) = \frac{Count(Y = y)}{\sum_{y'} Count(Y = y')}$$

  - 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: reconocimiento de dígitos

- Input: pixel grids



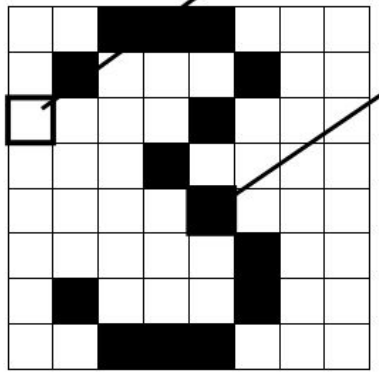
- Output: a digit 0-9

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 probabilísticos: regresión logística

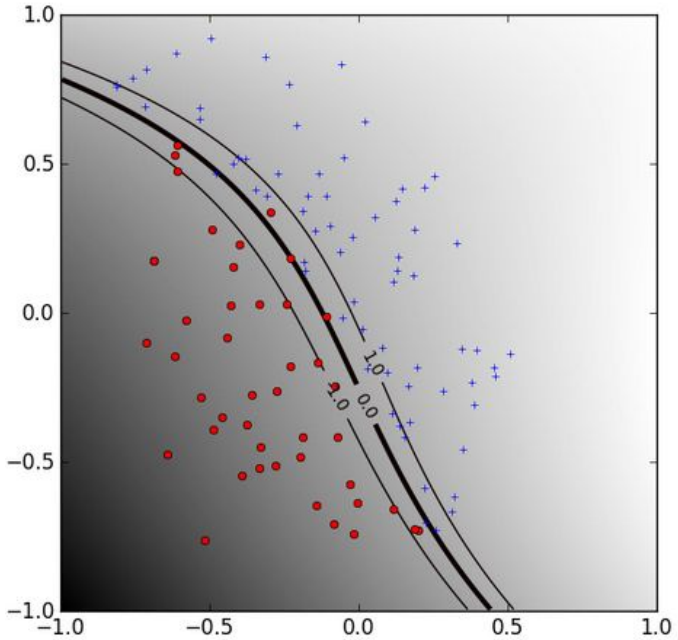
Clasificación basada en probabilidades

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

- Recordar:

$$0 \leq \mathrm{p}(evento) \leq 1$$

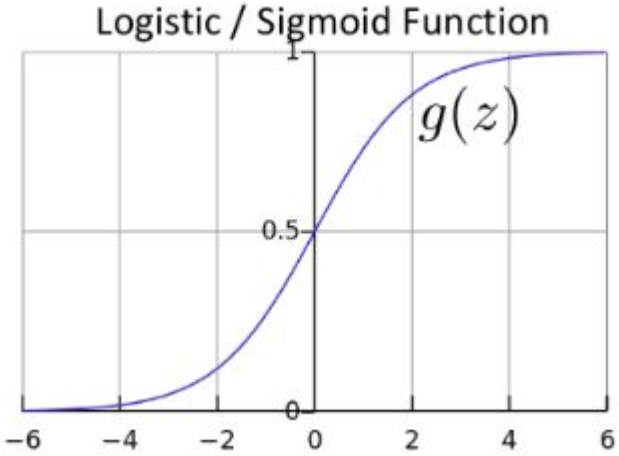
$$\mathrm{p}(evento) + \mathrm{p}(\neg evento) = 1$$



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

- 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

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

- Encontrar el $\boldsymbol{w}$ que maximice la verosimilitud de las etiquetas en el conjunto de entrenamiento

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

Regresión lineal vs. regresión logística