TEMA 1

CONCEPTOS DE PROBABILIDAD

DOMINIO: conjunto de todos los posibles eventos de un experimento aleatorio.

VARIABLE ALEATORIA: variable que define el experimento

SUCESO / EVENTO: valor que toma la variable al realizar el experimento.

PRINCIPIO INCLUSIÓN-EXCLUSIÓN

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

REGLA DE LA SUMA

$$P(A) + P(\bar{A}) = 1$$

REGLA DEL PRODUCTO

$$P(A,B) = P(A|B) \cdot P(B) \qquad P(A|B) = \frac{P(A,B)}{P(B)}$$

REGLA DE BAYES

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

porque $P(A,B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$

$$P(H|D,I) = \frac{P(D|H,I) \cdot P(H|I)}{P(D|I)}$$

$P(H|I)$: probabilidad a priori antes de ver los datos

$P(D|H,I)$: verosimilitud. Suponiendo H cierta como de verosimil es D.

$P(H|D,I)$: probabilidad a posteriori. La probabilidad de H "a la luz" de D

$P(D|I)$: evidencia. Probabilidad de los datos.

Toma de decisiones

Queremos escoger H^* entre $H_1, H_2, \cdots$

① Maximizar la prob. a posteriori (MAP)

$$H^* = \underset{H_i}{\operatorname{argmax}} P(H_i|D) = \underset{H_i}{\operatorname{argmax}} \frac{P(D|H_i) \cdot P(H_i)}{P(D)} =$$

$$= \underset{H_i}{\operatorname{argmax}} P(D|H_i) P(H_i)$$

② Máxima verosimilitud (MV)

$$H^* = \underset{H_i}{\operatorname{argmax}} P(D|H_i)$$

<u>Obs</u>: La ② es más fácil de calcular pero la ① da los mejores resultados.

Regla de Bayes simétrica

El mejor clasificador posible es el que clasifica cada ejemplo con la clase con mayor prob. a posteriori (MAP)
reduce prob. error

Regla asimétrica de Bayes

trucada justa

- Suponemos un coste R si pasamos una t por j.
- Suponemos un coste R' si pasamos una j por t.

Si observamos $D = c$ (cara), ¿qué coste tienen las decisiones...?

$$\begin{cases} ① \ h(c) = j \Rightarrow \mathbb{P}(t|c) \cdot R \\ ② \ h(c) = t \Rightarrow \mathbb{P}(j|c) \cdot R' \end{cases}$$

Si $D = c$:

$$h(c) = \begin{cases} j & \text{si} \quad \mathbb{P}(t|c) R < \mathbb{P}(j|c) \cdot R' \\ t & \text{si} \quad \mathbb{P}(t|c) R \geq \mathbb{P}(j|c) \cdot R' \end{cases}$$

$$\mathbb{P}(t|c) + \mathbb{P}(j|c) = 1$$

$$\mathbb{P}(j|c) > \frac{R}{R'}(1 - \mathbb{P}(j|c)) \Rightarrow \mathbb{P}(j|c) > \frac{R}{R+R'}$$

En conclusión:

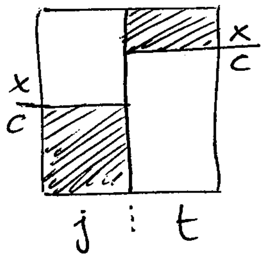
Si tenemos dos hipótesis H_1 y H_2, y al elegir H_1 por error tenemos un coste R y si al elegir H_2 por error tenemos un coste R' $\Rightarrow$ hay que elegir:

$$\begin{cases} H_1 \text{ si } \mathbb{P}(H_1|D) > \frac{R}{R+R'} \\ H_2 \text{ si } \mathbb{P}(H_2|D) > \frac{R'}{R+R'} \end{cases}$$

Continuamos con el ejemplo de las monedas: $\boxed{R=3R'}$

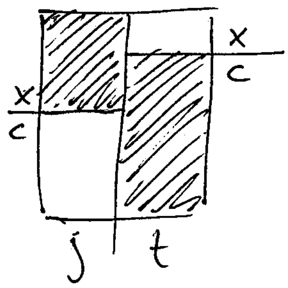
¿Modelo con menor coste? ¿ h_1, h_2, h_3 ó h_4?

$\boxed{h_1}$: $\begin{cases} h_1(c) = t \\ h_1(x) = j \end{cases}$



$$\text{coste}(h_1) = \mathbb{P}(j,c).R' + \mathbb{P}(t,x).R = \boxed{0'625.R'}$$

$\boxed{h_2}$: $\begin{cases} h_2(c) = j \\ h_2(x) = t \end{cases}$



$$\text{coste}(h_2) = \mathbb{P}(j,c).0 + \mathbb{P}(j,x).R' + \mathbb{P}(t,c).R + \mathbb{P}(t,x).0 =$$

$$= \mathbb{P}(t,c)R + \mathbb{P}(j,x)R' = 0'75.0'5.3R' + 0'5.0'5R' =$$

$$= \frac{9}{8}R' + \frac{1}{4}R' = \boxed{1'375\,R'}$$

$\boxed{h_3}$: $\text{coste}(h_3) = \boxed{1'5R'}$

$\boxed{h_4}$: $\text{coste}(h_4) = \boxed{0'5R'}$

Ejercicio: problema multidimensional (lentillas)

$H =$ no, duras, blandas

$D =$ edad (E), lesión (L), astigmatismo (A), prod. lágrimas (P)

) Llega un paciente con $L=m$, $P=n$. ¿$H=n$, $H=d$, $H=b$?

$$\mathbb{P}(H=n \mid L=m, P=n) = \frac{\mathbb{P}(L=m, P=n \mid H=n) \cdot \mathbb{P}(H=n)}{\mathbb{P}(L=m, P=n)} = (*)$$

sacamos $\mathbb{P}(\cdot)$ a partir de los datos

$$\boxed{\begin{array}{ll} \mathbb{P}(H=n) = \frac{15}{24} & \mathbb{P}(L=m, P=n \mid H=n) = \frac{1}{15} \\ \mathbb{P}(H=d) = \frac{4}{24} & \mathbb{P}(L=m, P=n \mid H=d) = \frac{3}{4} \\ \mathbb{P}(H=b) = \frac{5}{24} & \mathbb{P}(L=m, P=n \mid H=b) = \frac{2}{5} \end{array}}$$

$$(*) = \frac{1/15 \cdot 15/24}{\mathbb{P}(L=m, P=n)} = \frac{1/24}{\mathbb{P}(\dots)}$$

$$\mathbb{P}(H=d \mid L=m, P=n) = \frac{3/4 \cdot 4/24}{\mathbb{P}(L=m, P=n)} = \frac{3/24}{\mathbb{P}(\dots)} \Leftarrow$$

MAYOR PROB. A POSTERIORI

$$\mathbb{P}(H=b \mid L=m, P=n) = \frac{\qquad}{\mathbb{P}(L=m, P=n)} = \frac{2/24}{\mathbb{P}(\dots)}$$

Normalizando:

$$\mathbb{P}(H=n \mid L=m, P=n) = \frac{1}{6} \; ; \quad \mathbb{P}(H=d \mid L=m, P=n) = \frac{1}{2} \; ; \quad \mathbb{P}(H=b \mid L=m, P=n) = \frac{1}{3}$$

b) $L=h$, $P=n$	$1/3$	$1/6$	$1/2$
c) $L=m$, $P=r$	1	0	0
d) $L=h$, $P=r$	1	0	0
	$\mathbb{P}(H=n\mid\cdots)$	$\mathbb{P}(H=d\mid\cdots)$	$\mathbb{P}(H=b\mid\cdots)$

MATERIAL PARCIAL 1

Concepto (Maldición de la dimensionalidad): el nº de puntos necesarios para cubrir el espacio de atributos de forma uniforme crece exponencialmente con la dimensión.

Posible solución: NAÏVE-BAYES ("Bayes inocente")

Supondremos que los atributos son independientes dada la clase

Dado un vector de atributos $\vec{x} = (x_1, \dots, x_d)$, queremos saber la H^* más probable de entre las posibles. Es decir, debemos calcular $\mathbb{P}(H_i \mid \vec{x})$ $\forall i = 1, \dots, k$. ← k hipótesis $\{H_1, \dots, H_k\}$

aplicando la regla de Bayes:

$$\mathbb{P}(H_i \mid x_1, \dots, x_d) = \frac{\mathbb{P}(x_1, \dots, x_d \mid H_i) \cdot \mathbb{P}(H_i)}{\mathbb{P}(x_1, \dots, x_d)}$$

El cálculo de la verosimilitud requiere una cantidad de datos exponencial con la dimensión.

NAÏVE-BAYES
$$\mathbb{P}(H_i \mid x_1, \dots, x_d) = \frac{\mathbb{P}(x_1 \mid H_i)\,\mathbb{P}(x_2 \mid H_i) \cdots \mathbb{P}(x_d \mid H_i) \cdot \mathbb{P}(H_i)}{\mathbb{P}(x_1 \cdots x_2)}$$

$$= \frac{\prod_{j=1}^{d} \mathbb{P}(x_j \mid H_i) \cdot \mathbb{P}(H_i)}{\sum_{i}^{k} \prod_{j}^{d} \mathbb{P}(x_j \mid H_i)\,\mathbb{P}(H_i)}$$

MODELOS GRÁFICOS (REDES DE BAYES)

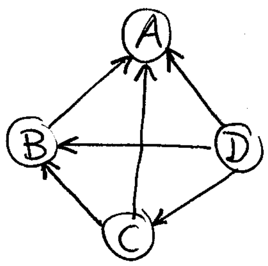
Permiten crear modelos que tengan en cuenta relaciones entre atributos

Regla de la cadena: aplicar la regla del producto múltiples veces

Podemos representar gráficamente con un grafo acíclico dirigido

Nodos: variables

Aristas: relaciones



La representación depende de como se aplique la regla de la cadena.

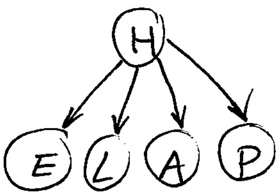
En general, $\mathbb{P}(x_1, \ldots, x_d) = \prod_{j=1}^{d} \mathbb{P}(x_j \mid pa_j)$

donde $pa_j \equiv$ conjunto de variables padre de x_j

si x_j no tiene progenitores $\mathbb{P}(x_j \mid pa_j) = \mathbb{P}(x_j)$

Ejemplo: modelo gráfico para NB y las lentillas

$$\mathbb{P}(E,L,A,P,H) = \mathbb{P}(E,L,A,P \mid H) \cdot \mathbb{P}(H) \overset{NB}{=} \mathbb{P}(E \mid H) \cdot \mathbb{P}(L \mid H) \cdot \mathbb{P}(A \mid H) \cdot \mathbb{P}(P \mid H) \cdot \mathbb{P}(H)$$



Regla de decisión de NB:

$$H^* = \underset{H_i}{\arg\max} \prod_{j=1}^{d} \mathbb{P}(x_j|H_i)\,\mathbb{P}(H_i)$$

Las $\mathbb{P}(x_j|H_i)$ y $\mathbb{P}(H_i)$ se estiman de los datos.

El entrenamiento consiste en crear tablas de cuentas de clases con respecto a los valores. Una tabla por atributo.

Por ejemplo, para el ejercicio de las lentillas

LESIÓN	no	d	b
miopia	7	3	2
hipermet	8	1	3

PRODUCCIÓN	no	d	b
normal	3	4	5
reducida	12	0	0

Solución para evitar ceros: CORRECCIÓN DE LAPLACE

Consiste en sumar 1 a todas las entradas de la tabla si esta contiene algún cero.

Obs: para atributos continuos las tablas pasaran a ser el cálculo de la media y varianza del atributo de cada clase

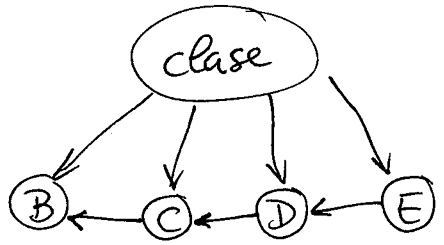
Coste computacional de Naive-Bayes

N: nº de ejemplos
D: nº de atributos
K: nº de clases

Coste de entrenamiento: $O(DN)$
Coste de clasificación: $O(DK)$

Ejemplo: Bayes gráfico

$$\mathbb{P}(clase, B, C, D, E) = \mathbb{P}(B \mid C, clase) \cdot \mathbb{P}(C \mid E, clase) \cdot \mathbb{P}(E \mid D) \cdot \mathbb{P}(D \mid clase) \cdot \mathbb{P}(clase)$$



$$\text{Bayes} \longrightarrow \mathbb{P}(clase \mid x_1, \dots, x_n) = \frac{\mathbb{P}(x_1, \dots, x_n \mid clase) \cdot \mathbb{P}(clase)}{\mathbb{P}(x_1, \dots, x_n)} =$$

$$= \frac{\mathbb{P}(clase, x_1, \dots, x_n)}{\mathbb{P}(x_1, \dots, x_n)} \quad \mathbb{P}(clase, x_1, \dots, x_n) \xrightarrow{\text{equiv.}} \mathbb{P}(clase, B, C, D, E)$$

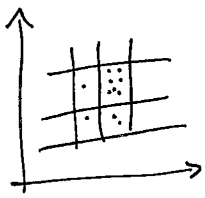
TEMA 2

MÉTODO DE VECINOS MÁS PRÓXIMOS

Introducción & Preliminares

Para atributos no discretos → opc. 1: discretización en intervalos/dominios
→ opc. 2: ajustar distr. de prob. paramétrica

Generalización de histogramas a más dimensiones usando la PDF:



2D $\quad \mathbb{P}(\bar{x} \in \text{Región}) = \int_{\text{Región}} PDF(\bar{x}) \, d\bar{x}$

Las estimaciones a la hora de discretizar son de dos formas:

① Por histogramas:

$$\mathbb{P}(\bar{x} \in \text{Región}) \simeq \frac{K_R}{N}$$ con K_R igual al nº de ptos. de la región R.

N debe ser grande para cubrir el espacio.
$\Longrightarrow$ Maldición de dimensionalidad

② Si suponemos que la región es pequeña, podemos suponer que la PDF es constante en dicha región.

$$\mathbb{P}(\bar{x} \in \text{Región}) = \int_{\text{Región}} PDF(\bar{x})\,d\bar{x} \simeq PDF(\bar{x}_R) \int_{\text{Región}} d\bar{x} = PDF(\bar{X}_R)\, V_R$$

$\bar{X}_R \to$ centro región
$V_R \to$ volumen región

__Bonus__: Si ahora igualamos ① y ②: $\boxed{PDF(\bar{x}_R) \simeq \frac{K_R}{N \cdot V_R}}$

Esta fórmula se suele usar de dos formas:

- Fijar el volumen V_R y calcular K $\Longrightarrow$ MÉTODO DE NUCLEOS
- Fijar K y V_R se define para que caigan K ptos. dentro $\Longrightarrow$
$\Longrightarrow$ MÉTODO DE VECINOS PRÓXIMOS

__Bayes__: Verosimilitud para la clase $K \longrightarrow \mathbb{P}(\bar{x}|C_K) \simeq \frac{K_R}{N.V_R}$

Prob. a priori $\longrightarrow \mathbb{P}(C_K) = \frac{N_K}{N}$

Evidencia $\longrightarrow \mathbb{P}(\bar{x}) = \frac{K}{N.V_R}$

$$\mathbb{P}(C_K|\bar{x}) = \frac{\frac{K_R}{\cancel{N_K}.V_R} \cdot \frac{\cancel{N_K}}{N}}{\frac{K}{N.V_R}} = \frac{\frac{K_R}{N.V_R}}{\frac{K}{N.V_R}} = \frac{K_R}{K}$$

¿Qué clase maximiza $\mathbb{P}(C_K|\bar{x})$? Aquella que tiene más instancias en V_R $\Longrightarrow$ VECINOS MÁS PRÓXIMOS (K-NN)

Algoritmo de entrenamiento <u>K-NN</u> $\longrightarrow$ coste $= O(D.N)$ ← si normalizamos

- Cosas previas importantes que hay que hacer:
  1. Decidir métrica/distancia (euclídea, Manhattan, ...)
  2. Normalizar datos: para cada atributo se calcula media y desviación típica: $\frac{\bar{x}-\mu}{\sigma} = \bar{x}\_new$

     Se guardan los valores de la media y la desv. std.

     En scikit-learn: StandardScaler()

Algoritmo de clasificación <u>K-NN</u> $\longrightarrow$ coste $= O(DN)$

- Algoritmo de K-NN: normalizamos los datos, calculamos distancias a todos los ptos. y nos quedamos con los K más cercanos. Devolvemos la moda de las clases en K.

MODELOS LINEALES

[...]

Si $\mathbb{P}(C_1) = \mathbb{P}(C_2)$ y $\Sigma_1 = \Sigma_2 = \mathbb{I}$ entonces:

$$\text{Si } (\bar{x}-\mu_2)^2 > (\bar{x}-\mu_1)^2 \Longrightarrow C_1$$

$$\text{si } (\bar{x}-\mu_2)^2 \leq (\bar{x}-\mu_1)^2 \Longrightarrow C_2$$

Es decir, $\bar{x}$ se clasifica como C_1 si está más cerca de la media de los ejemplos de la clase 1.

<u>Obs</u>: ¿Qué frontera de decisión tenemos? (A partir de ahora $\mathbb{P}(C_1)$ no tiene por qué ser igual a $\mathbb{P}(C_2)$)

$$\mathbb{P}(C_1|\bar{x}) = 0'5 \Longrightarrow \sigma(a) = 0'5 \Longrightarrow a = 0$$

$$a = \frac{(\bar{x}-\mu_2)^2}{2} - \frac{(\bar{x}-\mu_1)^2}{2} + \ln\left(\frac{\mathbb{P}(C_1)}{\mathbb{P}(C_2)}\right) = 0$$

$$\boxed{\underbrace{(\mu_1-\mu_2)}_{w = \mu_1 - \mu_2}\bar{x} + \underbrace{\frac{\mu_2^2-\mu_1^2}{2} + \ln\left(\frac{\mathbb{P}(C_1)}{\mathbb{P}(C_2)}\right)}_{w_0 = b = \frac{\mu_2^2-\mu_1^2}{2} + \ln\left(\frac{\mathbb{P}(C_1)}{\mathbb{P}(C_2)}\right)}}$$ ECUACIÓN DE UN HIPERPLANO

$w_0 = b$ ← dependiendo del libro

- Si $\mathbb{P}(C_1) = \mathbb{P}(C_2) \Longrightarrow$ Mediatriz de μ_1 y μ_2
- Si $\mathbb{P}(C_1) > \mathbb{P}(C_2) \Longrightarrow$ la frontera se aleja de μ_1 hacia μ_2.

Si ahora suponemos:

- 2 clases: C_1 y C_2
- $\mathbb{P}(\bar{x}|C_1) = N(\mu_1, \Sigma)$
- $\mathbb{P}(\bar{x}|C_2) = N(\mu_2, \Sigma)$

$\Big\}$ Es decir, $\Sigma_1 = \Sigma_2 = \Sigma$

llegaríamos a que $\omega = \Sigma^{-1}(\mu_1 - \mu_2)$

$$b = \omega_0 = \frac{1}{2}\mu_1 \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2 \Sigma^{-1}\mu_2 + \ln \frac{\mathbb{P}(C_1)}{\mathbb{P}(C_2)}$$

Para entrenar un modelo lineal podríamos estimar $\mu_1, \mu_2, \Sigma, \mathbb{P}(C_1)$ y $\mathbb{P}(C_2)$ y con eso sacamos ω y ω_0. ¿Cuántos parámetros son?

$$\underset{\mu_1}{D} + \underset{\mu_2}{D} + \underset{\Sigma}{\frac{D(D+1)}{2}} + \underset{\mathbb{P}(C_1)}{1} + \underset{\mathbb{P}(C_2)}{1}$$

Realmente solo necesitamos saber ω y ω_0: <u>$D+1$ parámetros</u>

Vamos a redefinir los vectores para simplificar notación:

$$\omega = (\omega_0, \underbrace{\omega_1, \dots, \omega_d}_{\text{anterior } \omega})$$

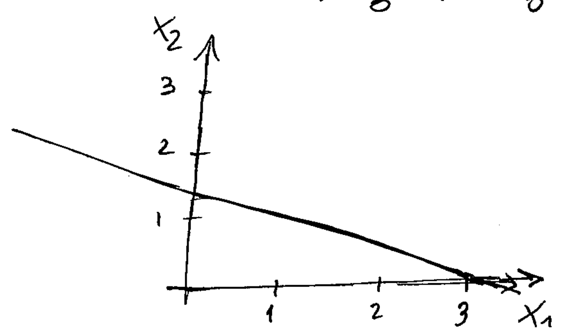
$$\bar{x} = (1, \underbrace{x_1, \dots, x_d}_{\text{anterior } x})$$

de esta forma $\mathbb{P}(C_1|\bar{x})$ queda $\boxed{\mathbb{P}(C_1|x) = \sigma(\omega x)}$

<u>Ejemplo</u>: $\omega = (-1, 1/3, 2/3)$

¿frontera? $\sigma(\omega x) = \frac{1}{2} \Rightarrow \omega x = 0 \Rightarrow \langle(-1, 1/3, 2/3), (1, x_1, x_2)\rangle = 0 \Rightarrow$

$$\Rightarrow \frac{1}{3}x_1 + \frac{2}{3}x_2 - 1 = 0$$



- frontera de decisión

Tenemos un conjunto de N ejemplos: $\mathcal{D} = \{\{\bar{x}_j, t_j\}, j = 1, \dots, N\}$ donde $t_j = 1$ si $\bar{x}_j$ es C_1, ó $t_j = 0$ si $\bar{x}_j$ es C_2.

Vamos a definir el modelo de REGRESIÓN LOGÍSTICA.

TEMA 3

REGRESIÓN LOGÍSTICA

Se estima $\bar{\omega}$ por MV: $\mathbb{P}(t_1, \dots, t_N | \bar{\omega}, \bar{x}_1, \dots, \bar{x}_N) =$

$$= \prod_{j=1}^{N} \mathbb{P}(t_j | \bar{\omega}, \bar{x}_j) \qquad \text{donde } \mathbb{P}(t_j | \bar{\omega}, \bar{x}_j) = \begin{cases} \sigma(\bar{\omega}\bar{x}_j) & \text{si } t_j = 1 \\ 1 - \sigma(\bar{\omega}\bar{x}_j) & \text{si } t_j = 0 \end{cases}$$

Podemos escribir: $\mathbb{P}(t_j | \bar{\omega}, \bar{x}_j) = \sigma(\bar{\omega}\bar{x}_j)^{t_j} \cdot (1 - \sigma(\bar{\omega}\bar{x}_j))^{1-t_j}$

$$\Longrightarrow \mathbb{P}(t_1, \dots, t_N | \bar{\omega}, \bar{x}_1, \dots, \bar{x}_N) = \prod_{j=1}^{N} \mathbb{P}(t_j | \bar{\omega}, \bar{x}_j) = \prod_{j=1}^{N} \sigma(\bar{\omega}.\bar{x}_j)^{t_j} \cdot (1 - \sigma(\bar{\omega}.\bar{x}_j))^{1-t_j}$$

Cómo queremos calcular $\bar{\omega}$ que maximice lo anterior, derivamos:

Antes observar: $\prod_{j=1}^{N} \sigma(\bar{\omega}.\bar{x}_j)^{t_j} \cdot (1 - \sigma(\bar{\omega}.\bar{x}_j))^{1-t_j} = \sum_{j=1}^{N} t_j \log(\sigma(\bar{\omega}\bar{x}_j)) + (1-t_j)\log(1 - \sigma(\bar{\omega}\bar{x}_j))$

Ahora minimizamos $- \sum_{j=1}^{N} t_j \cdot \log(\sigma(\bar{\omega}\bar{x}_j)) + (1-t_j)\log(1 - \sigma(\bar{\omega}\bar{x}_j)) = E$

$$\frac{\partial E}{\partial \bar{\omega}} = - \sum_{j=1}^{N} \left[t_j \frac{1}{\sigma(\bar{\omega}\bar{x}_j)} - (1-t_j) \frac{1}{1 - \sigma(\bar{\omega}\bar{x}_j)} \right] \frac{\partial \sigma(\bar{\omega}\bar{x}_j)}{\partial \bar{\omega}} =$$

$$= - \sum_{j=1}^{N} \left[t_j \frac{1}{\sigma(\bar{\omega}\bar{x}_j)} - (1-t_j) \frac{1}{1 - \sigma(\bar{\omega}\bar{x}_j)} \right] \sigma(\bar{\omega}\bar{x}_j)(1 - \sigma(\bar{\omega}\bar{x}_j)) \bar{x}_j =$$

$$= - \sum_{j=1}^{N} \left[t_j (1 - \sigma(\bar{\omega}\bar{x}_j)) - (1-t_j)\sigma(\bar{\omega}\bar{x}_j) \right] \bar{x}_j =$$

$$= \sum_{j=1}^{N} (\sigma(\bar{\omega}\bar{x}_j) - t_j) \bar{x}_j = 0$$

Para mejorar la verosimilitud del ejemplo $\bar{x}_j$ hay que mover la recta en sentido opuesto al gradiente $(\sigma_j - t_j)\bar{x}_j$ y proporcional a una constante de aprendizaje η: $\boxed{\bar{\omega} = \bar{\omega} - \eta(\sigma_j - t_j)\bar{x}_j}$

Algoritmo de regresión logística:

▷ Entrenamiento: parámetros η, nepocas

- Generamos $\bar{\omega}$ aleatorio con coefs. en $[-0'5, 0'5]$.
- Para ep desde 0 a nepocas:

    Para j desde 1 a N:

    $\sigma = \sigma(\bar{\omega}\bar{x}_j)$ # prob. a posteriori de C_1

    $\bar{\omega} = \bar{\omega} - \eta(\sigma - t_j)\bar{x}_j$

  return $\bar{\omega}$

VERSIÓN MAP DE REG. LOGÍSTICA

Verosimilitud:

$$\mathbb{P}(t_1, \dots, t_N \mid \bar{\omega}, \bar{x}_1, \dots, \bar{x}_N) = \prod_{j=1}^{N} \sigma(\bar{\omega}x_j)^{t_j} (1 - \sigma(\bar{\omega}x_j))^{1-t_j}$$

Suponemos que el módulo de $|\bar{\omega}|$ viene dado por una gaussiana de media 0

$$\mathbb{P}(|\bar{\omega}|) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-|\bar{\omega}|^2/2\sigma^2}$$

$\boxed{\text{OBS: } \sigma^2 \equiv \text{varianza},\quad \sigma_j \equiv \sigma(\bar{\omega}x_j)}$

Prob. a posteriori

$$\mathbb{P}(\bar{\omega} \mid t_1, \dots, t_N, \bar{x}_1, \dots, \bar{x}_N) \simeq \underbrace{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-|\omega|^2/2\sigma^2}}_{\text{prior}} \underbrace{\prod_{j=1}^{N} \sigma_j^{t_j}(1-\sigma_j)^{1-t_j}}_{\text{verosimilitud}}$$

Aplicando logaritmos:

$$E = \ln \mathbb{P}(\bar{\omega} \mid t_1, t_2, \dots, \bar{x}_1, \dots, \bar{x}_N)$$

$$\frac{\partial E}{\partial \bar{\omega}} = 0 \iff \cdots \iff \frac{\partial E}{\partial \bar{\omega}} = \sum_{j=1}^{N} (\sigma_j - t_j)\bar{x}_j + \frac{\bar{\omega}}{N\sigma^2}$$

Algoritmo reg. logística map:

reg_log_map(η, nepocas, σ^2):

- Generar $\bar{\omega}$ aleatoriamente con coefs. en $[-0'5, 0'5]$ $O(D)$
- For $i: 0 \to$ nepocas:

  For $j: 1 \to N$:

$$\bar{\omega} = \bar{\omega} - \eta(\sigma_j - t_j)\bar{x_j} - \frac{\eta\bar{\omega}}{N\sigma^2}$$

$O(\text{nepocas} \cdot N \cdot D)$

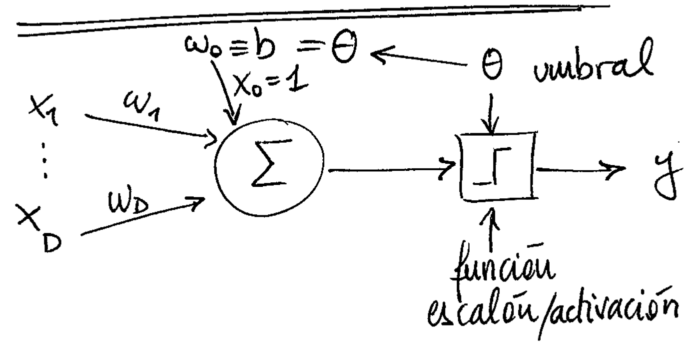
Coste de entrenamiento $\longrightarrow$ $O(\text{nepocas} \cdot N \cdot D)$

Coste de clasificación $\longrightarrow$ $O(D)$ (coste de realizar un producto escalar)

Observaciones finales:

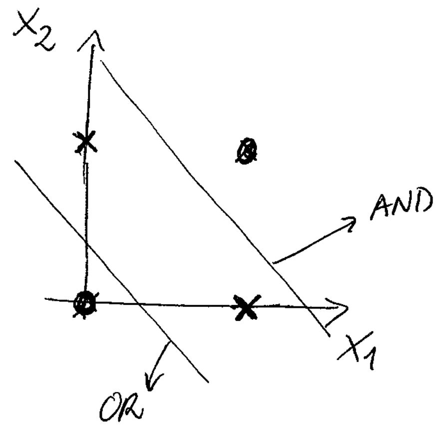
- RegLog trabaja solo con atributos numéricos
- Se recomienda normalizar atributos antes de entrenar

Redes neuronales

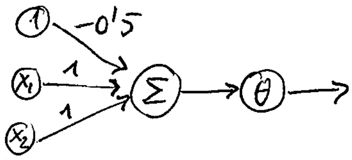


$$y(\bar{x}) = \begin{cases} 1, & \text{si } \bar{\omega}\bar{x} \geq 0 \\ 0, & \text{si } \bar{\omega}\bar{x} < 0 \end{cases}$$

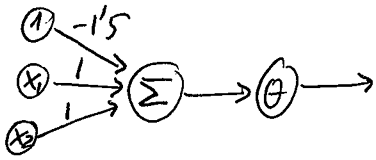
PERCEPTRÓN DE ROSENBLATT



$$z_1 = \theta(x_1 + x_2 - 0'5)$$



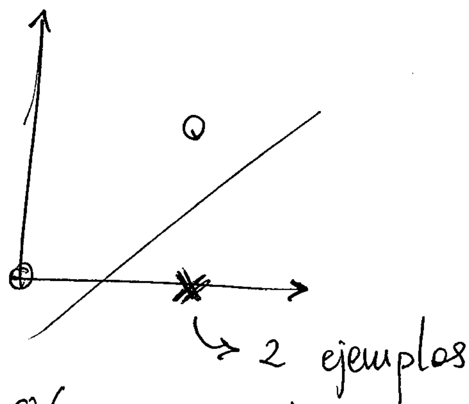
$$z_2 = \theta(x_1 + x_2 - 1'5)$$



$\theta :=$ función de Heaviside

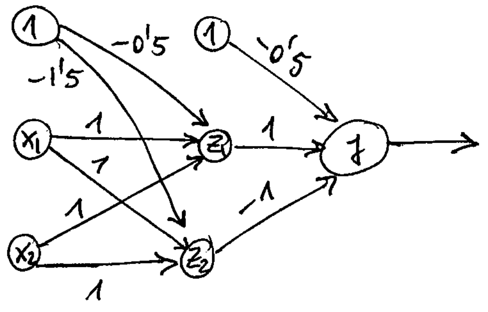
x_1	x_2	t	z_1	z_2
0	0	0	0	0
0	1	1	1	0
1	0	1	1	0
1	1	0	1	1

Hemos definido nuevos atributos $\longrightarrow$



$$y = \theta(z_1 - z_2 - 0'5)$$

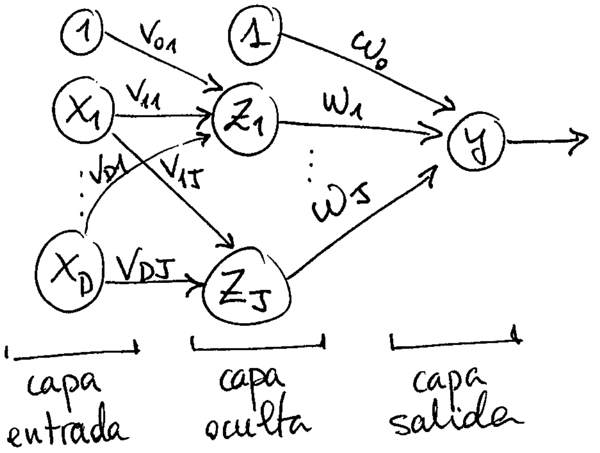
z_1	z_2	y
0	0	0
1	0	1
1	0	1
1	1	0



Regla delta

Sustituir $\underset{\text{f. escalón}}{\theta} \longrightarrow \underset{\text{f. sigmoidal}}{\sigma}$

Red general de 3 capas



no. parámetros: $(D+1)J + J + 1$

v_{DJ} := peso entre la neurona D (entrada) y la J (oculta)

w_j := peso entre neurona j (oculta) y la capa de salida

$$z_j = \sigma\left(\sum_{d=0}^{D} x_d v_{dj}\right)$$

$$y(\bar{x}) = \sigma\left(\sum_{j=0}^{J} w_j z_j\right) = \sigma\left(\sum_{j=0}^{J} w_j \sum_{d=0}^{D} x_d v_{dj}\right)$$

$$\mathcal{D} = \{(\bar{x}_n, t_n)\}_{n=1}^{N} \quad \text{(conjunto de datos)}$$

$$\mathbb{P}(t_1, \dots, t_N \mid v_{00} \_\_ v_{DJ}, w_0, \_\_, w_J, \bar{x}_1, \_, \bar{x}_N) =$$

$$= \prod_{n=1}^{N} \mathbb{P}(C_1 \mid \bar{x}_n, v, \_\_ w, \_)^{t_n} \left(1 - \mathbb{P}(C_1 \mid \bar{x}_n, v\_, w, \_)\right)^{1-t_n} =$$

$$= \prod_{n=1}^{N} y(\bar{x}_n)^{t_n} (1 - y(\bar{x}_n))^{1-t_n}$$

Aplicando logaritmos:

$$E(v_0, \_\_, w_0, \_) = -\log \mathbb{P}(t_1 \_ t_N \mid v \_ w \_ \bar{x} \_ \bar{x}_N) =$$

$$= \sum_{n=1}^{N} E_n(v \_ w) \quad \text{con} \quad E_n = -t_n \log y(\bar{x}_n) - (1-t_n) \log(1 - y(\bar{x}_n))$$

Vamos a derivar lo anterior.

Derivada comodín: $\frac{\partial E_n}{\partial \beta} = -\frac{t_n}{y(\bar{x}_n)} \cdot \frac{\partial y(\bar{x}_n)}{\partial \beta} + \frac{1-t_n}{1-y(\bar{x}_n)} \cdot \frac{\partial y(\bar{x}_n)}{\partial \beta} = (\star)$

$$\frac{\partial y(\bar{x}_n)}{\partial \beta} = \frac{\partial}{\partial \beta} \cdot \frac{1}{1+e^{-wz}} = y(\bar{x}_n).(1-y(\bar{x}_n)).\frac{\partial wz}{\partial \beta}$$

$$\Rightarrow (\star) = -\frac{t_n}{y(\bar{x}_n)} \cdot \frac{\partial y(\bar{x}_n)}{\partial \beta} + \frac{1-t_n}{1-y(\bar{x}_n)} \cdot y(\bar{x}_n)(1-y(\bar{x}_n)) \frac{\partial wz}{\partial \beta} \Rightarrow$$

$$\Rightarrow (\star) = \left[-t_n(1-y(\bar{x}_n)) + (1-t_n)y(\bar{x}_n)\right]\frac{\partial wz}{\partial \beta} = \left(y(\bar{x}_n) - t_n\right)\frac{\partial wz}{\partial \beta}$$

- Derivamos con respecto a w_i: (renombramos $\beta \to w_i$)

$$\frac{\partial wz}{\partial w_i} = \frac{\partial}{\partial w_i}(w_0 + w_1 z_1 + \cdots + w_J z_J) = z_i$$

$$\Rightarrow \boxed{\frac{\partial E_n}{\partial w_i} = \left(y(\bar{x}_n) - t_n\right) z_i}$$

- Derivamos con respecto a V_{pq}: (renombramos $\beta \to V_{pq}$):

$$\frac{\partial wz}{\partial V_{pq}} = \frac{\partial}{\partial V_{pq}} \sum_{j=0}^{J} w_j z_j = \sum_{j=0}^{J} w_j \frac{\partial}{\partial V_{pq}} \sigma\left(\sum_{j=0}^{D} x_{nd} V_{dj}\right) =$$

$$= \sum_{j=0}^{J} w_j z_j (1-z_j) \frac{\partial}{\partial V_{pq}} \sum_{j=0}^{D} x_{nd} V_{dj} = \sum_{j=0}^{J} w_j z_j (1-z_j) \sum_{d=0}^{D} x_{nd} \frac{\partial V_{dj}}{\partial V_{pq}} =$$

$$= \sum_{j=0}^{J} w_j z_j (1-z_j) \sum_{j=0}^{D} x_{nd} \underbrace{\delta_{dp}\,\delta_{jq}}_{\text{delta de Kronecker}} = \sum_{j=0}^{J} w_j z_j (1-z_j) x_{np}\,\delta_{jq} =$$

$$= w_q z_q (1-z_q) x_{np}$$

$$\Rightarrow \boxed{\frac{\partial E_n}{\partial V_{pq}} = \left(y(\bar{x}_n) - t_n\right) w_q z_q (1-z_q) x_{np}}$$

Reglas de actualización:

$$w_i = w_i - \eta\,(y(\bar{x}_n) - t_n)\,z_i$$

$$V_{pq} = V_{pq} - \eta\,(y(\bar{x}_n) - t_n)\,w_q\, z_q\,(1 - z_q)\,x_{np}$$

Pseudocódigo

NN_model (η, nepocas):

inicializar pesos de la red aleatoriamente $\in [-0'5, 0'5]$.

for ie = 1 : nepocas

for n = 1 : N

$$z_q = \sigma\left(\sum_{d=0}^{D} x_{nd} V_{dq}\right) \rightarrow O(D.J)$$

$$y = \sigma\left(\sum_{j=0}^{J} w_j z_j\right) \rightarrow O(J)$$

forward prop.

$$V_{pq} = V_{pq} - \eta\,(y - t_n)\,w_q\, z_q\,(1 - z_q)\,x_{np} \rightarrow O(DJ)$$

$$w_q = w_q - \eta\,(y - t_n)\,z_q \rightarrow O(J)$$

backward prop.

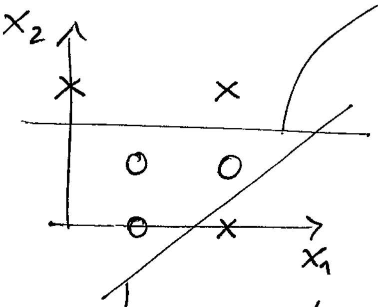
Coste de entrenamiento: $O(\text{nepocas} . N . D . J)$

Coste de clasificación: $O(D.J)$

Ejemplo 1:

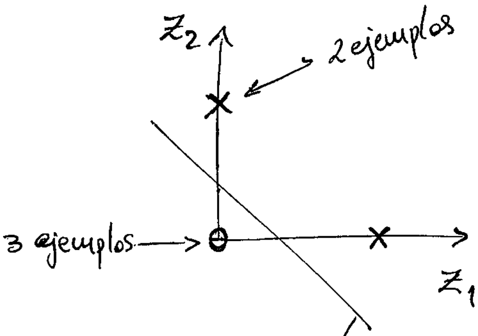
x_1	x_2	t
0'5	0	0
1	0	1
1	0'5	0
0	1	1
0'5	0'5	0
1	1	1

$x_2 = \theta(x_2 - 0'75)$



Vamos a usar funciones escalón

$z_1 = \theta(x_1 - x_2 - 0'75)$



$y = \theta(z_1 + z_2 - 0'5)$

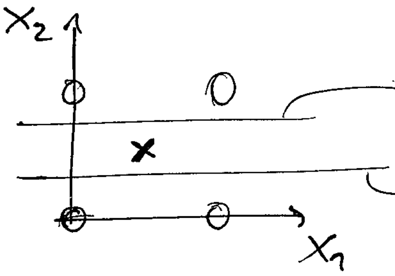
cuidado con este signo en la última capa

$\Downarrow$

x_1	x_2	t	z_1	z_2
0'5	0	0	0	0
1	0	1	1	0
1	0'5	0	0	0
0	1	1	0	1
0'5	0'5	0	0	0
1	1	1	0	1

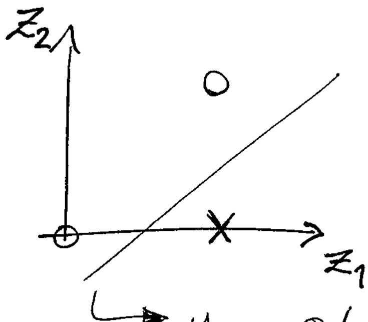
Ejemplo 2

x_1	x_2	t	z_1	z_2
0	0	0	0	0
0	1	0	1	1
1	0	0	0	0
1	1	0	1	1
0'5	0'5	1	1	0



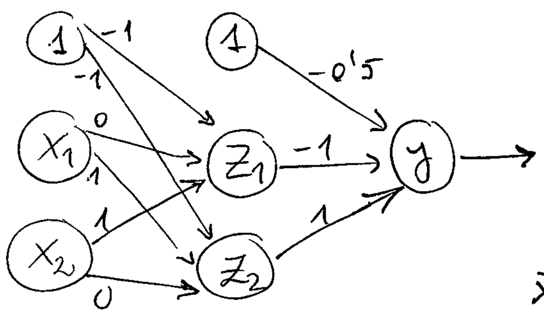
$z_2 = \theta(x_2 - 0'75)$

$z_1 = \theta(x_2 - 0'25$

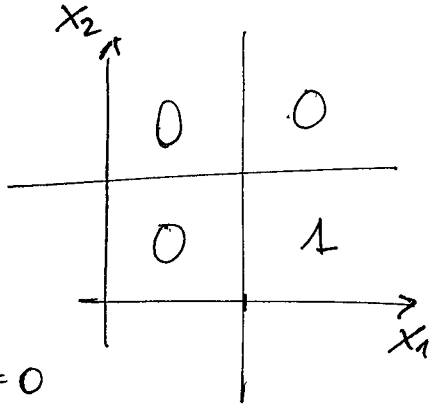


$y = \theta(z_2 - z_1 - 0'5)$

Ejemplo 3



$$z_1 = \theta(x_2 - 1)$$
$$z_2 = \theta(x_1 - 1)$$
$$y = \theta(z_2 - z_1 - 0'5)$$

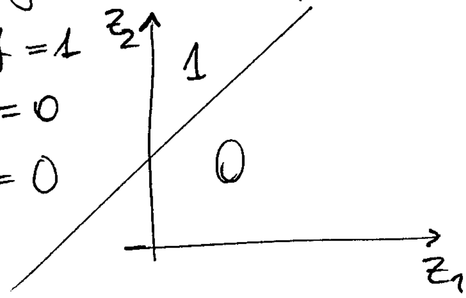


$\vec{x} = (0,0) \Rightarrow \vec{z} = (0,0) \Rightarrow y = 0$

$\vec{x} = (2,0) \Rightarrow \vec{z} = (0,1) \Rightarrow y = 1$

$\vec{x} = (0,2) \Rightarrow \vec{z} = (1,0) \Rightarrow y = 0$

$\vec{x} = (2,2) \Rightarrow \vec{z} = (1,1) \Rightarrow y = 0$



Pregunta: ¿Qué pasaría en estos ejemplos si
(examen) la func. de activación es la sigmoidal σ?

[TEMA 5]

ALGORITMOS GENÉTICOS

Los algoritmos genéticos utilizan las ideas de la evolución para resolver problemas de optimización.

- problema
- cada individuo codifica una solución
- función de ajuste (fitness)
- conjunto de soluciones
- mecanismos aleatorios de adaptación
  - → los individuos con mayor fitness tienen más probabilidad de reproducirse
  - → modificaciones aleatorias en las soluciones (mutaciones)

Algoritmo genético genérico

```
ag(fitness f, params de evolución):
    Init población inicial aleatoria P
    Mientras no se cumpla criterio_fin:
        S = selección_progenitores(P)
        S = recombinación(S)
        S = mutación(S)
        P = selección_supervivientes(S,P)
    return best(P)
```

¿Por qué funcionan los AG? TEORÍA DE ESQUEMAS

Los algoritmos genéticos funcionan descubriendo y recombinando buenos bloques de soluciones en paralelo.

La teoría de esquemas formaliza el concepto de bloque:

- La codificación binaria un bloque o esquema es una cadena de 0,1 y $*$, donde $*$ representa 0 ó 1.

  $H = 1***0 \rightarrow$ todas las soluciones de longitud 5 que empiezan por 1 y acaban por 0.

- Orden de un esquema H: $O(H) = \#$ bits definidos
- Longitud de un esquema H: $d(H) =$ distancia entre los bits definidos más lejanos.

  $d(0***1) = 4$

  $d(01*) = 1$

  $d(0**) = 0$

- Número de cadenas binarias posibles de long. L: 2^L
- Número de cadenas con 1,0,$*$ posibles de long. L: 3^L.
- Cada cadena definida de longitud L es una instancia de 2^L esquemas: la cadena 101 pertenece a 2^3 esquemas.
- Una población de N individuos contiene instancias de entre 2^L esquemas (si son todos los individuos iguales) a $N2^L$ esquemas
- A pesar de estar evaluando solo N individuos, estos en realidad implícitamente ajustan más esquemas.

Notación

$n_H(t) \equiv$ número de instancias del esquema H en tiempo t.

$f_i(t) \equiv$ fitness individuo i en tiempo t.

$\bar{f}(t) \equiv$ fitness medio de la población en tiempo t.

$\bar{f}_H(t) \equiv$ fitness medio del esquema H en tiempo t.

Queremos obtener el valor esperado de instancias en $t+1$ para H: $\mathbb{E}[n_H(t+1$

Paso 1: Selección proporcional a fitness

$$\mathbb{E}_s[n_i(t+1)] = \frac{f_i(t)}{\sum_i^N f_i(t)} \cdot N = \frac{f_i(t)}{\bar{f}(t)}$$

$$\mathbb{E}_s[n_H(t+1)] = \frac{1}{\bar{f}(t)} \sum_{i \in H} f_i(t) = n_H(t) \cdot \frac{\bar{f}_H(t)}{\bar{f}(t)}$$

Si $\bar{f}_H(t) > \bar{f}(t) \Rightarrow \mathbb{E}_s[n_H(t+1)] > n_H(t)$ en media

Paso 2: Cruce en un punto

Suponemos P_c de cruce. Tenemos que determinar si un esquema sobrevive al cruce. Decimos que sobrevive si al menos un descendiente pertenece al esquema H. Una cota inferior es:

$$S_c = 1 - P_c \cdot \frac{d(H)}{L-1}$$

Cuanto mayor sea $d(H)$ menor probabilidad de supervivencia tiene.

Paso 3: Mutación bitflip

Una instancia sobrevive si no se cambia ningún bit definido en H. $S_M = (1 - P_M)^{O(H)}$

Finalmente: $$\mathbb{E}[n_H(t+1)] = \frac{\bar{f}_H(t)}{\bar{f}(t)} n_H(t) \cdot \left(1 - P_c \frac{d(H)}{L-1}\right) \cdot (1 - P_M)^{O(H)}$$

ÁRBOLES DE DECISIÓN

PROBLEMA DE LA SUMA (NP)

Lista de números: 3, 8, 14, 2, 25, 5, 7 (long. K)
Número objetivo: 15

Hay que seleccionar los números de la lista que sumen el número objetivo.

Queremos solucionar el problema utilizando un algoritmo genético. Necesitamos codificar las soluciones, cruce y mutación.

- Representación:
  - $\xrightarrow{\text{opc. 1}}$ lista binaria de longitud K
    $[1,0,0,0,1,0,0] \equiv 3+25=28$
  - $\xrightarrow{\text{opc. 2}}$ lista de enteros de longitud $\leq$ K.
    $[3, 25] \equiv 3+25=28$

- Fitness:
  - $\xrightarrow{\text{opc. 1}}$ $f(x) = \dfrac{1}{|obj - Suma| + 1}$
  - $\xrightarrow{\text{opc. 2}}$ $f(x) = -|obj - Suma|$

- Cruce:
  - $\xrightarrow{\text{opc. 1}}$ cruce en un punto
    $\begin{matrix} 1\,0\,0\,1 & | & 0\,0 \\ 0\,1\,1\,0 & | & 1\,0 \end{matrix} \rightarrow \begin{cases} 100110 \\ 011000 \end{cases}$
  - $\xrightarrow{\text{opc. 2}}$ cruce en un punto con eliminación de repeticiones
    $\begin{matrix} 3 & | & 7\;\;8 \\ 2 & | & 3\;\;5 \end{matrix} \rightarrow \begin{cases} 3\;\;5 \\ 2\;\;7\;\;8 \end{cases}$
  - $\xrightarrow{\text{opc. 1 mejorada}}$ cruce uniforme
    $\begin{matrix} c\,x\,x\,x\,c \\ \hline 0\,0\,1\,1\,0 \\ 1\,0\,1\,0\,1 \end{matrix} \rightarrow \begin{matrix} 0\,0\,1\,0\,0 \\ 1\,0\,1\,1\,1 \end{matrix}$
    c = arriba
    x = abajo

- Mutación $\xrightarrow{\text{opc. 1}}$ bit-flip

  $\xrightarrow{\text{opc. 2}}$ añadir/quitar 1 número