Chuleta-2do-PARCIAL.pdf

user_3208335

Sistemas Inteligentes

3º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingeniería Informática
Universidad Politécnica de Valencia

Ω --> **Espacio muestral**, dentro de una muestra puede tener diferentes características (**variable aleatoria**), cada característica puede tener diferentes valores (**tiempo**: **despejado o nublado**)

	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}
Ω	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
	des dia seg	nub dia seg	nub noc acc	llu noc acc	nub dia seg	des noc seg	llu dia seg	des dia seg	nub noc seg	des dia acc

La suma de todas las probabilidades de una característica tiene que dar 1

P(des) + P(nub) + P(llu) = 1

P(dia) + P(noc) = 1

P(seg) + P(acc) = 1

Probabilidad a priori --> incondicional
Probabilidad conjunta --> dos variables a la vez
Probabilidad condicional --> probabilidad de un valor sobre el total de otro valor fijo

P(llu | dia) = 1/6

regla producto: $P(x,y) = P(x)\,P(y \mid x) = P(y)\,P(x \mid y)$

Dos **variables** son **independientes** si **aplicando la regla del producto** es el **producto de las variables** --> $P(x)\,P(y)$ ó si la **prob. condicional no cambia**

Regla de Bayes:

$$P(y \mid x) = \frac{P(x,y)}{P(x)} = \frac{P(y)\,P(x \mid y)}{P(x)} = \frac{P(y)\,P(x \mid y)}{\sum_{y'} P(y')\,P(x \mid y')}$$

Probabilidad a posteriori — Verosimilitud

$$P(\text{gripe} \mid 39) = \frac{P(39 \mid \text{gripe})\,P(\text{gripe})}{P(39)}$$

Para tomar decisiones, hay que minimizar el riesgo de error (1 - prob.)

REPRESENTATION VECTORIAL DE LAS MUESTRAS — VECTOR DE CARACTERÍSTICAS

	muestra 1	muestra 2	muestra 3	muestra 4	muestra 5
long. pétalo	5.1	4.9	7.0	3.7	6.5
anch. pétalo	3.4	3.0	3.2	3.3	2.4
long. sépalo	1.4	1.4	4.7	6.0	5.3
anch. sépalo	0.2	0.6	1.4	2.5	1.8

Si tenemos **cuatro características --> R^4**, dependiendo de los valores, se pueden clasificar en diferentes clases
Funciones lineales discriminantes:

$$y = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

pesos

variables discriminantes = características de las muestras

término independiente

$(w0 \;\; w1 \;\; w2 \ldots wn)$ vector pesos (función lineal)

$\begin{pmatrix} x1 \\ x2 \\ . \\ . \\ xn \end{pmatrix}$ vector de características de las muestras

Con el vector de pesos, introduciendo los valores de las variables podemos averiguar a qué clase pertenece cada muestra

$g_A(y) > g_B(y)$ --> clase A
$g_A(y) < g_B(y)$ --> clase B
$g_A(y) = g_B(y)$ --> frontera

Frontera de decisión --> Se consigue igualando las lineas discriminantes, sirve para separar las regiones de cada clase
Clasificadores equivalentes --> definen misma frontera y mismas regiones
Se puede emplear esta formula para calcular:

G' = G · a + b — **> 0**

$$\begin{matrix}(1 & -1 & 1) = (0 & -1 & 1) \cdot a + b \\ (1 & 1 & -1) = (0 & 1 & -1) \cdot a + b\end{matrix}$$

Ejercicio **Perceptrón**
Pesos iniaciales nulo, debe estar todo en anotación homogenea (se suele añadir el valor 1 arriba de los vectores de la muestra)

y1 = (1, 0, 0) — y2 = (1, 1, 1)
alpha = 1 — b = 0,1
Factor aprendizaje — Margen

$y_1 = \begin{pmatrix}0\\0\end{pmatrix}$ c=1, $y_2 = \begin{pmatrix}1\\1\end{pmatrix}$ c=2, $w_1 = (0\;0\;0)$, $w_2 = (0\;0\;0)$

Iteración 1:
n (muestra) = 1 — g = w1 * y1 = 0 (Porque w1 es el vector nulo)
g2 = w2 * y1 + b = 0,1 (Provoca un ERROR, g2 > g)
Como se ha originado un error, hay que actualizar los valores de las w:
w2 = w2 - alpha * y1 = (-1, 0, 0)
w1 = w1 + alpha * y1 = (1, 0, 0)

CUIDADO!!! HAY QUE EMPLEAR LOS VALORES NUEVOS CADA VEZ QUE SE ACTUALIZAN, TAMBIÉN FIJARSE CON LOS VECTORES DE PESO

n = 2 — g = w2 * y2 = -1 (Usando el valor actualizado)
g2 = w1 * y2 + b = 1,1 (Provoca un ERROR, g2 > g)
Como se ha originado un error, hay que actualizar los valores de las w:
w1 = w1 - alpha * y2 = (0, -1, -1)
w2 = w2 + alpha * y2 = (0, 1, 1)

REPETIR HASTA ENCONTRAR UNA ITERACIÓN SIN ERRORES

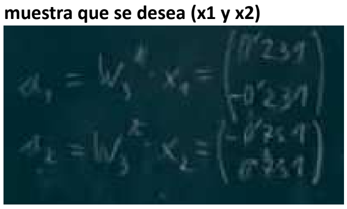
Ejercicio **Regresión Logística**
X = {(1, 0, 0), (1, 1, 1)} — Y = {(1, 0), (0, 1)}
Factor aprendizaje = 1 — Matriz de pesos iniciales nulos (W0)
Softmax, aplicar para todos los componentes : epsilon elevado un componente del vector resultado W0^t * x dividido entre el sumatorio de epsilon evelado a cada componente

Iteración 1:
Softmax1 = (W0 ^ t) * x1 = (0, 0) --> µ1 = ((e^0)/((e^0) + (e^0)), (e^0)/((e^0) + (e^0))) = **(0,5, 0,5)**
Softmax2 = (W0 ^ t) * x2 = (0, 0) --> µ2 = ((e^0)/((e^0) + (e^0)), (e^0)/((e^0) + (e^0))) = **(0,5, 0,5)**

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Aplicar **descenso por gradiente**:
W1 = W0 - factor de aprendizaje * (1/N) * Sumatorio (cada muestra * (µn - Yn)^t)
N es el número de muestras
Haciendo **operaciones dentro** del **sumatorio para la primera muestra**:
(1, 0, 0)^t * ((0,5, 0,5)^t - (1, 0)^t) = (1, 0, 0)^t * (-0,5, 0,5) = (-0,5 0,5)
Habría que hacer esto para cada uno de las muestras (0 0)
y después realizar el sumatorio (0 0)
El **resultado del sumatorio** sería:
(0 0) — Para calcular W1 = (0 0)
(0,5 -0,5) — (0 0) - 1 * sumatorio / N
(0,5 -0,5) — (0 0)
FIN DE LA PRIMERA ITERACIÓN, AHORA HAY QUE REPETIRE EL PROCESO HASTA LA ITERACIÓN REQUERIDA UTILIZANDO LA NUEVA W1, W2, W3…

Después de realizar las iteraciones se nos pide calcular la probabilidad a posteriori, hay que coger la última W y hacer lo siguiente:
Realiza el **cálculo de a1 y a2**, que son la **multiplicación** de la **última W y** la **muestra que se desea (x1 y x2)**



Posteriormente, hay que aplicar la softmax para terminar este apartado

Verosimilitud

Sumatorio de todas las muestras (Sumatorio de todas las clases(Ync * log(µnc)))

$$\mathrm{LL}(\mathbf{W}) = \sum_{n=1}^{N}\sum_{c=1}^{C} y_{nc}\log\mu_{nc}$$

µ es la función SOFTMAX (W * x = epsilon partido de epsilon….). Después hay que aplicar a dicho valor: Ycn * log (µnc)

Ejemplo: log-verosimilitud de $\mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix}$ con dos datos $\mathcal{D} = \{((1,0,0)^t,(1,0)^t),((1,1,1)^t,(0,1)^t)\}$

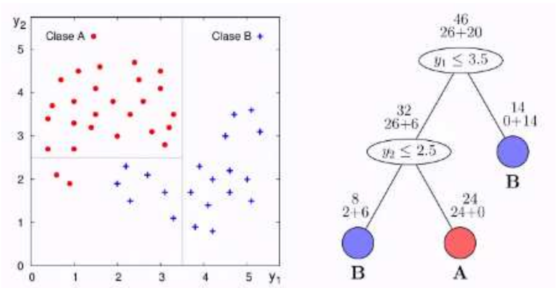
$$\begin{aligned}\mathrm{LL}(\mathbf{W}) &= \sum_{n=1}^{N}\log\prod_{c=1}^{C}\mu_{nc}^{y_{nc}} = \sum_{n=1}^{N}\sum_{c=1}^{C} y_{nc}\log\mu_{nc}\\ &= y_{11}\log\mu_{11} + y_{12}\log\mu_{12} + y_{21}\log\mu_{21} + y_{22}\log\mu_{22}\\ &= \log\mu_{11} + \log\mu_{22}\\ &= \log 0.8808 + \log 0.8808 = -0.1269 - 0.1269 = -0.2538\end{aligned}$$

Los primeros componentes de D son los datos y los segundos son Y

NLL = -1/N * LL, hay que buscar el mínimo valor NLL

Descenso por gradiente: Theta i+1 = Theta i - factor de aprendizaje * derivada de Theta sobre Theta i

Si Theta^2 --> Derivada = 2*Theta, sustituyendo en la función sería **Ti+1 = Ti - FA * 2*T0**



Se puede decir que la **rama derecha** de la primera partición es **pura** ya que **no hay ningún dato de la otra clase**

- N número de muestras de entrenamiento
- N_c número de muestras de entrenamiento que pertenecen a la clase c
- $N(t)$ número de muestras de entrenamiento representadas en un nodo t
- $N_c(t)$ número de muestras de entrenamiento de un nodo t que pertenecen a la clase c

Probabilidad a priori de la clase c:	$\hat{P}(c) = \frac{N_c}{N}$
Probabilidad a posteriori de la clase c en el nodo t:	$\hat{P}(c \mid t) = \frac{N_c(t)}{N(t)}$
Probabilidad de un nodo terminal $t \in \tilde{T}$:	$\hat{P}(t) = \frac{N(t)}{N}$
Probabilidad de seleccionar el hijo izquierdo de t:	$\hat{P}_t(L) = \frac{N(t_L)}{N(t)}$
Probabilidad de seleccionar el hijo derecho de t:	$\hat{P}_t(R) = \frac{N(t_R)}{N(t)}$

Nodo terminal, probabilidad de que termine en un nodo (t)

Splits, 1 sola componente(j) y 1 valor límite(r) --> s(j,r,t). j puede tomar valores desde 1 hasta D (grado de dimensión), r = N (num. muestras) + 1

Impureza, - Sumatorio de todas las clases(prob. a posteriori * log en base 2(prob. a posteriori))

Es mejor cuando la **impureza del nodo padre > impureza de los nodos hijos directos** (teniendo en cuenta el porcentaje de las ramas)

Entropía, cantidad de información asociada a una decisión k-aria

$$H = -\sum_{i=1}^{k} P_i \log_2 P_i \qquad (0\log 0 \overset{\text{def}}{=} 0)$$

- Ejemplos:
  - Si $P_1 = P_2 = 1/2$, $\quad H = -(0.5(0-1) + 0.5(0-1)) = 1$ bit
  - Si $P_1 = 1,\ P_2 = 0$, $\quad H = -1 \cdot 0 + 0 = 0$ bits
  - Si $P_i = 1/k,\ 1 \le i \le k$, $\quad H = \log_2 k;\ H \to \infty$ si $k \to \infty$

Error de estimación, Sumatorio ((nodos de la región / nodos totales) * pocerntaje de nodos mal clasificados)

Suma de Errores Cuadráticos (SEC), sólo son apropiado para clusters esféricos de tamaño similar

Dada una partición de N datos en C *clusters* $\Pi = \{X_1,\dots,X_C\}$, su SEC es:

$$J(X_1,\dots,X_C) = \sum_c J_c\,,\quad J_c = \sum_{x\in X_c}\|x - m_c\|^2,\quad m_c = \frac{1}{|X_c|}\sum_{x\in X_c} x \qquad (2)$$

mc es **media** de cada cluster, y también es el “prototipo natural”

Se quiere reducir el SEC

Algoritmo K-medias / C-medias

$$\triangle J = \underbrace{\frac{n_j}{n_j+1}\|x - m_j\|^2}_{\text{clúster destino de } x} - \underbrace{\frac{n_i}{n_i-1}\|x - m_i\|^2}_{\text{clúster origen de } x}$$

$$\Delta J = \frac{4}{4+1}\|x - m_j\|^2 - \frac{3}{3-1}\|x - m_i\|^2$$

La transferencia es favorable si el incremento de SEC es negativo

$$m_i' = m_i - \frac{x - m_i}{n_i - 1} \qquad m_j' = m_j + \frac{x - m_j}{n_j + 1}$$

Si es favorable, se mueve la muestra y hay que calcular la nueva media

m' es la nueva media, m es la vieja media, x es valor de muestra y n es el número de muestras que había en el cluster. Las variables i son del origen y j de destino