Sistemas Inteligentes

Escuela Técnica Superior de Informática

Universitat Politècnica de València

Tema B2T2

Regresión logística

SIN

Índice

Índice

Codificación one-hot y distribución categórica

- ***Variable categórica:*** variable aleatoria que toma un valor de un conjunto finito de categorías (no ordenadas)
- ***Ejemplos de variables categóricas:*** color RGB, ***etiqueta de clase***, palabra de un vocabulario, etc.
- ***Codificación one-hot:*** de una variable categórica y que toma un valor entre C posibles, $\{1, \dots, C\}$

$$\text{one-hot}(y) = \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_C \end{pmatrix} = \begin{pmatrix} \mathbb{I}(y=1) \\ \vdots \\ \mathbb{I}(y=C) \end{pmatrix} \in \{0,1\}^C \quad \text{con} \quad \sum_c y_c = 1$$

donde

$$\mathbb{I}(x) = \begin{cases} 1 & \text{si } x \text{ es Verdadero} \\ 0 & \text{en caso contrario} \end{cases}$$

Codificación one-hot y distribución categórica

- ***Distribución categórica:*** distribución de probabilidades entre las C posibles categorías de una variable categórica, que viene dada por un vector de parámetros $\boldsymbol{\theta} \in [0,1]^C$ tal que $\sum_c \theta_c = 1$

$$p(y \mid \boldsymbol{\theta}) = \operatorname{Cat}(y \mid \boldsymbol{\theta}) = \prod_{c=1}^{C} \theta_c^{\mathbb{I}(y=c)}$$

  o en notación one-hot,

$$p(y \mid \boldsymbol{\theta}) = \operatorname{Cat}(\boldsymbol{y} \mid \boldsymbol{\theta}) = \prod_{c=1}^{C} \theta_c^{y_c}$$

- ***Convención:*** $0^0 = 1$ y $0 \log 0 = 0$
- ***Ejemplo:*** $\boldsymbol{\theta} = (0.5, 0.5, 0)^t$, $\operatorname{Cat}(\boldsymbol{y} = (1,0,0)^t \mid \boldsymbol{\theta}) = 0.5^1 0.5^0 0^0 = 0.5$

Modelo probabilístico de clasificación con softmax

- ***Normalización probabilística de clasificadores:*** todo clasificador G definido con funciones discriminantes generales $[g_1, \cdots, g_C]$ puede representarse mediante un clasificador equivalente G' con funciones discriminantes normalizadas probabilísticamente $[g'_1, \cdots, g'_C]$

$$\begin{aligned} c(\boldsymbol{x}) &= \underset{c}{\operatorname{argmax}}\ g_c(\boldsymbol{x}) \\ &= \underset{c}{\operatorname{argmax}}\ e^{g_c(\boldsymbol{x})} && \text{con } h(z) = e^z \in \mathbb{R}^{\geq 0} \text{ estrictamente creciente} \\ &= \underset{c}{\operatorname{argmax}}\ \frac{e^{g_c(\boldsymbol{x})}}{\sum_{c'} e^{g_{c'}(\boldsymbol{x})}} && \text{con } h(z) = kz,\ k \text{ constante positiva (invariable con } c) \end{aligned}$$

  Por tanto, $g'_c(\boldsymbol{x}) = \frac{e^{g_c(\boldsymbol{x})}}{\sum_{\tilde{c}} e^{g_{\tilde{c}}(\boldsymbol{x})}}$ define un clasificador equivalente. Esta transformación de g_c a g'_c es más conocida como ***función softmax***.

- En este modelo probabilístico se asume que los valores $g_c(\boldsymbol{x})$ son log-probabilidades no normalizadas denominados ***logits***.

Modelo probabilístico de clasificación con softmax

- ***La función softmax:*** transforma un vector de ***logits*** (log-probabilidades no normalizadas) $G \in \mathbb{R}^C$ en uno de probabilidades $G' \in [0,1]^C$

$$G' = \mathcal{S}(G) = \left[\frac{e^{g_1}}{\sum_{\tilde{c}} e^{g_{\tilde{c}}}}, \ldots, \frac{e^{g_C}}{\sum_{\tilde{c}} e^{g_{\tilde{c}}}}\right]$$

donde se cumple

$$0 \leq \mathcal{S}(G)_c \leq 1 \quad \text{y} \quad \sum_c \mathcal{S}(G)_c = 1$$

Modelo probabilístico de clasificación con softmax

- ***Modelo probabilístico de clasificación con softmax:*** se predicen las probabilidades de todas las clases a partir de un clasificador $G : \mathcal{X} \rightarrow \mathbb{R}^C$

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathrm{Cat}(\boldsymbol{y} \mid \mathcal{S}(G)) = \prod_c (\mathcal{S}(G))_c)^{y_c}$$

- ***Conveniencia del modelo en inferencia:*** la predicción de las probabilidades de todas las clases permite aplicar reglas de clasificación más generales que la clasificación por máxima probabilidad a posteriori. Por ejemplo, para la aplicación de funciones de error que son diferentes para cada clase.

- ***Conveniencia del modelo en aprendizaje:*** permite plantear el aprendizaje probabilísticamente, con criterios estándar como máxima verosimilitud; además, gracias a la softmax, G se puede elegir libremente sin estar sujetos a las restricciones de la probabilidad.

Regresión logística

- ***Regresión logística:*** modelo con softmax y funciones discriminantes lineales (en notación homogénea)

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{W}) = \mathrm{Cat}(\boldsymbol{y} \mid \boldsymbol{\mu})$$

donde

$$\boldsymbol{\mu} = \mathcal{S}(\boldsymbol{a}), \quad \boldsymbol{a} = f(\boldsymbol{x}; \mathbf{W}) = \mathbf{W}^t\boldsymbol{x}, \quad \mathbf{W} \in \mathbb{R}^{D\times C} \quad \text{y} \quad \boldsymbol{x} \in \mathbb{R}^D$$

- No hay diferencia con los clasificadores basados en funciones discriminantes lineales, a excepción de que ahora predecimos las probabilidades de todas las clases

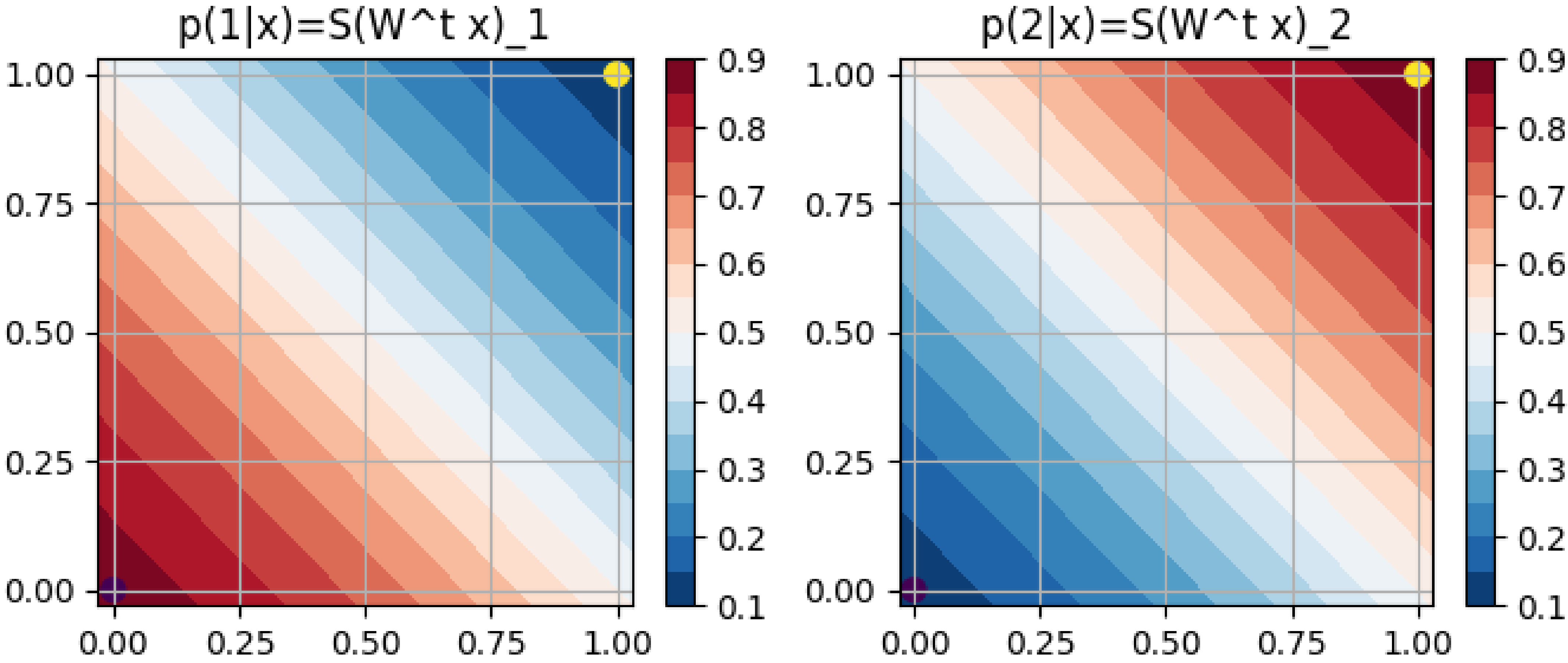
Ejemplo de regresión logística

$$C = D = 2, \quad a_1 = g_1(x_1, x_2) = -x_1 - x_2 + 1, \quad a_2 = g_2(x_1, x_2) = x_1 + x_2 - 1$$

$$\boldsymbol{a} = f(\boldsymbol{x}; \mathbf{W}) = \mathbf{W}^t \boldsymbol{x} \quad \text{con} \quad \mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix} \quad \text{y} \quad \boldsymbol{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$$

$\boldsymbol{x}^t$	$\boldsymbol{a}^t$	$\mu_1 = \mathcal{S}(\boldsymbol{a})_1$	$\mu_2 = \mathcal{S}(\boldsymbol{a})_2$
$(1, 0, 0)$	$(1, -1)$	$\frac{e^1}{e^1+e^{-1}} = 0.8808$	$\frac{e^{-1}}{e^1+e^{-1}} = 0.1192$
$(1, 1, 1)$	$(-1, 1)$	$\frac{e^{-1}}{e^{-1}+e^1} = 0.1192$	$\frac{e^1}{e^{-1}+e^1} = 0.8808$
$(1, 0.5, 0.5)$	$(0, 0)$	$\frac{e^0}{e^0+e^0} = 0.5000$	$\frac{e^0}{e^0+e^0} = 0.5000$

Ejemplo de regresión logística



Aprendizaje por máxima verosimilitud

- ***Objetivo:*** establecer un criterio para aprender $\mathbf{W}$ a partir de un conjunto de datos de entrenamiento, $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$
- ***Log-verosimilitud (condicional):*** log-probabilidad de $\mathcal{D}$ interpretada como función de $\mathbf{W}$

$$\begin{aligned}
\mathrm{LL}(\mathbf{W}) = \log p(\mathcal{D} \mid \mathbf{W}) &= \log \prod_{n=1}^{N} p(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \mathbf{W}) \\
&= \sum_{n=1}^{N} \log \mathrm{Cat}(\boldsymbol{y}_n \mid \boldsymbol{\mu}_n) \quad \text{con} \quad \boldsymbol{\mu}_n = \mathcal{S}(\boldsymbol{a}_n) \quad \text{y} \quad \boldsymbol{a}_n = \mathbf{W}^t \boldsymbol{x}_n \\
&= \sum_{n=1}^{N} \log \prod_{c=1}^{C} \mu_{nc}^{y_{nc}} \\
&= \sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log \mu_{nc}
\end{aligned}$$

Aprendizaje por máxima verosimilitud

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

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

- ***Aprendizaje por máxima verosimilitud:*** elegimos una $\mathbf{W}$ que otorgue máxima probabilidad a $\mathcal{D}$

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmax}} \ \mathrm{LL}(\mathbf{W})$$

Planteamiento como un problema de minimización

- ***Neg-log-verosimilitud:*** log-verosimilitud con el signo cambiado y normalizada por el número de datos

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

con

$$\boldsymbol{\mu}_n = \mathcal{S}(\boldsymbol{a}_n) \quad \text{y} \quad \boldsymbol{a}_n = \mathbf{W}^t\boldsymbol{x}_n$$

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

$$\mathrm{NLL}(\mathbf{W}) = -\frac{1}{2}\mathrm{LL}(\mathbf{W}) = 0.1269$$

Planteamiento como un problema de minimización

- ***Riesgo empírico con log-pérdida:*** es lo mismo que NLL

$$\mathcal{L}(\mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) = \mathrm{NLL}(\mathbf{W})$$

con

$$\ell(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) = -\log p(\boldsymbol{y}_n \mid \boldsymbol{\mu}_n) = -\sum_{c=1}^{C} y_{nc} \log \mu_{nc}$$

  - Si el modelo asigna probabilidad uno a la clase correcta, la pérdida es nula
  - Si no, la pérdida será positiva y será más grande cuanto menor sea la probabilidad asignada a la clase correcta

- ***Aprendizaje por mínima NLL:*** aprendizaje por máxima verosimilitud planteado como un problema de minimización

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}}\ \mathrm{NLL}(\mathbf{W})$$

Planteamiento como un problema de minimización

- ***Ejemplo:*** $\mathcal{D} = \{((1,0,0)^t, (1,0)^t), ((1,1,1)^t, (0,1)^t)\}$; por simplicidad, suponemos que tenemos que elegir por mínima NLL entre

$$\mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix} \qquad \text{y} \qquad \tilde{\mathbf{W}}^t = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & -1 \end{pmatrix}$$

Elegimos $\mathbf{W}$ ya que su NLL, 0.1269 (ver antes), es menor que la de $\tilde{\mathbf{W}}$:

$$\mathrm{NLL}(\tilde{\mathbf{W}}) = -\frac{1}{2}(\log \tilde{\mu}_{11} + \log \tilde{\mu}_{22}) = -\log \frac{e^{-1}}{e^{-1} + e^{1}} = \log(1 + e^2) = 2.1269$$

Aprendizaje con descenso por gradiente

- ***Objetivo:*** A diferencia del riesgo con pérdida 01 (tasa de error en entrenamiento), el riesgo con log-pérdida es derivable, por lo que podemos minimizarlo con técnicas de optimización estándar como descenso por gradiente
- ***Descenso por gradiente:*** algoritmo iterativo para minimizar una función $\mathcal{L}(\boldsymbol{\theta})$ a partir de un valor inicial de los parámetros $\boldsymbol{\theta}_0$ dado

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \eta_i \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}_i}$$

- ***Factor de aprendizaje:*** $\eta_i > 0$ juega el mismo papel que en Perceptrón; podemos elegir un valor pequeño constante, $\eta_i = \eta$
- ***Dirección de descenso más pronunciada:*** $-\boldsymbol{\nabla} \mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}_i}$ es el neg-gradiente de la función evaluada en $\boldsymbol{\theta}_i$
- ***Convergencia:*** si η no es muy grande y la función es convexa (con forma de bol), converge a un mínimo (global)

Ejemplo de descenso por gradiente

- ***Ejemplo:*** $\mathcal{L}(\theta) = \theta^2, \ \theta_0 = 10, \ \eta_t = 0.2, \ \frac{d\mathcal{L}}{d\theta} = 2\theta$ y tolerancia 0.01

θ	$\mathcal{L}(\theta)$
-4.0	6.0
-2.4	3.6
-1.44	2.16
-0.864	1.296
-0.5184	0.7776
-0.311	0.4666
-0.1866	0.2799
-0.112	0.168
-0.0672	0.1008
-0.0403	0.0605
-0.0242	0.0363
-0.0145	0.0218
-0.0087	0.0131

Descenso por gradiente aplicado a regresión logística

- ***NLL:*** la NLL es una función convexa

$$\mathrm{NLL}(\mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N} -\log p(\boldsymbol{y}_n \mid \boldsymbol{\mu}_n) \qquad \text{con} \quad \boldsymbol{\mu}_n = \mathcal{S}(\boldsymbol{a}_n) \quad \text{y} \quad \boldsymbol{a}_n = \mathbf{W}^t \boldsymbol{x}_n$$

- ***Gradiente de la NLL:*** haremos uso del siguiente resultado, sin demostrar

$$\begin{pmatrix} \frac{\partial\,\mathrm{NLL}}{\partial W_{11}} & \cdots & \frac{\partial\,\mathrm{NLL}}{\partial W_{1C}} \\ \vdots & \ddots & \vdots \\ \frac{\partial\,\mathrm{NLL}}{\partial W_{D1}} & \cdots & \frac{\partial\,\mathrm{NLL}}{\partial W_{DC}} \end{pmatrix} = \frac{\partial\,\mathrm{NLL}}{\partial \mathbf{W}^t} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t$$

- ***Descenso por gradiente aplicado a regresión logística:***

$$\mathbf{W}_0 = \mathbf{0}; \qquad \mathbf{W}_{i+1} = \mathbf{W}_i - \eta_i \, \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t$$

Ejercicio de regresión logística

Dado un problema de clasificación en dos clases con dos datos bidimensionales $\mathcal{D} = \{((1,0,0)^t, (1,0)^t), ((1,1,1)^t, (0,1)^t)\}$:

- Realiza tres iteraciones del algoritmo de aprendizaje del modelo de regresión logística que minimiza la neg-log-verosimilitud con descenso por gradiente ($\eta = 1.0$) a partir de la matriz de pesos iniciales nulos.
- Calcula la probabilidad a posteriori de los datos a partir de la matriz de pesos final obtenida en el anterior apartado.
- Clasifica los datos por máxima probabilidad a posteriori.

Solución del ejercicio de regresión logística

i	$\frac{\partial\,\mathrm{NLL}}{\partial \mathbf{W}}$	$\mathbf{W}_{i+1}^t$
0	$\begin{pmatrix} 0.0 & 0.25 & 0.25 \\ 0.0 & -0.25 & -0.25 \end{pmatrix}$	$\begin{pmatrix} 0.0 & -0.25 & -0.25 \\ 0.0 & 0.25 & 0.25 \end{pmatrix}$
1	$\begin{pmatrix} -0.12 & 0.13 & 0.13 \\ 0.12 & -0.13 & -0.13 \end{pmatrix}$	$\begin{pmatrix} 0.12 & -0.38 & -0.38 \\ -0.12 & 0.38 & 0.38 \end{pmatrix}$
2	$\begin{pmatrix} -0.11 & 0.11 & 0.11 \\ 0.11 & -0.11 & -0.11 \end{pmatrix}$	$\begin{pmatrix} 0.23 & -0.49 & -0.49 \\ -0.23 & 0.49 & 0.49 \end{pmatrix}$

$$p(\boldsymbol{Y} \mid \boldsymbol{X}, \mathbf{W}) = \mathcal{S}(\boldsymbol{X}\mathbf{W}) = \begin{pmatrix} 0.61 & 0.39 \\ 0.18 & 0.82 \end{pmatrix} \text{ con } \boldsymbol{X} \in \mathbb{R}^{N\times D} \text{ y } \mathbf{W} \in \mathbb{R}^{D\times C}$$

Por máxima probabilidad a posteriori, el primer dato se clasifica en la primera clase y el segundo dato en la segunda clase.

Índice

Bibliografía

[1] Kevin P. Murphy. Probabilistic Machine Learning: An introduction. MIT Press, 2022.