Clasificación

Matias Vera - Juan Zuloaga- Lautaro Estienne

Centro de Simulación Computacional para Aplicaciones Tecnológicas

Agenda

Teoría de Clasificación

Bases

Objetivo: Clasificar Y (con $|\mathcal{Y}|$ finito) a partir del valor de X: $\hat{Y} = \varphi(X)$
Función costo: Hard $\rightarrow \quad \ell(x, y) = \mathbb{1}\left\{y \neq \varphi(x)\right\}$
Riesgo Esperado: Probabilidad de error $\rightarrow \quad \mathbb{P}\left(Y \neq \varphi(X)\right)$

Teoría de Clasificación

Bases

Objetivo: Clasificar Y (con $|\mathcal{Y}|$ finito) a partir del valor de X: $\hat{Y} = \varphi(X)$
Función costo: Hard $\rightarrow \quad \ell(x, y) = \mathbb{1}\{y \neq \varphi(x)\}$
Riesgo Esperado: Probabilidad de error $\rightarrow \quad \mathbb{P}(Y \neq \varphi(X))$

Optimalidad

$$\mathbb{P}(Y \neq \varphi(X)) \geq 1 - \mathbb{E}\left[\max_y P_{Y|X}(y|X)\right]$$

con igualdad si y solo si $\varphi(x) = \arg\max_y P_{Y|X}(y|x)$.

Clasificador Bayesiano: $\varphi(x) = \arg\max_y P_{Y|X}(y|x)$

Error Bayesiano: $1 - \mathbb{E}\left[\max_y P_{Y|X}(y|X)\right]$

Clasificador bayesiano

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{P}\left(Y \neq \varphi(X)\right)$. Es decir aprender el "clasificador bayesiano": $\varphi(x) = \arg\max_y P_{Y|X}(y|x)$.

Clasificador bayesiano

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{P}\left(Y \neq \varphi(X)\right)$. Es decir aprender el "clasificador bayesiano": $\varphi(x) = \arg\max_y P_{Y|X}(y|x)$.

Problemas numéricos

La propuesta de buscar $\varphi(\cdot)$ que minimice el riesgo empírico: $\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{Y_i \neq \varphi(X_i)\right\}$ suele tener problemas numéricos (no derivable).

Clasificador bayesiano

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{P}(Y \neq \varphi(X))$. Es decir aprender el "clasificador bayesiano": $\varphi(x) = \arg\max_y P_{Y|X}(y|x)$.

Problemas numéricos

La propuesta de buscar $\varphi(\cdot)$ que minimice el riesgo empírico: $\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{Y_i \neq \varphi(X_i)\}$ suele tener problemas numéricos (no derivable).

Posible solución

El clasificador bayesiano se aprenderá en dos etapas:

- Aprender toda $P_{Y|X}(y|x)$.
- Quedarse con el máximo.

Clasificadores extremos

Clasificador bayesiano

El mejor clasificador (en términos de la probabilidad de error) es:

$$\mathbb{P}(Y \neq \varphi(X)) \geq 1 - \mathbb{E}\left[\max_y P_{Y|X}(y|X)\right]$$

Clasificador al azar para k clases

Cualquier clasificador razonable debe ganarle a la decisión al azar:

$$\mathbb{P}(Y \neq \varphi(X)) \leq 1 - \frac{1}{k}$$

Clasificador dummy

Otro clasificador muy precario (pero mejor que el azaroso) es elegir siempre la clase más probable. La probabilidad de error del dummy es:

$$\mathbb{P}(Y \neq \varphi(X)) \leq 1 - \max_y P_Y(y)$$

Tarea

Sea $Y \sim \text{Ber}(3/4)$, $X|Y=0 \sim \mathcal{N}(0,4)$ y $X|Y=1 \sim \mathcal{N}(0,1)$. Calcular la $P_{Y|X}(y|x)$, el clasificador bayesiano y graficarlo sobre la distribución *mezcla* (conjunta). Además computar el error bayesiano, el error de un *clasificador al azar* y el error del *clasificador dummy*.

Divergencia de Kullback Leibler

Kullback Leibler

$$\mathrm{KL}(P\|Q) = \sum_{y\in\mathcal{Y}} P(y)\log\left(\frac{P(y)}{Q(y)}\right)$$

Teorema

$$\mathrm{KL}(P\|Q) \geq 0$$

con igualdad si y solo si $P(y) = Q(y)$ para todo $y \in \mathcal{Y}$.
(*Hint:* $\log(x) \leq x - 1$).

Divergencia de Kullback Leibler

Kullback Leibler

$$\mathrm{KL}(P\|Q) = \sum_{y\in\mathcal{Y}} P(y)\log\left(\frac{P(y)}{Q(y)}\right)$$

Teorema

$$\mathrm{KL}(P\|Q) \geq 0$$

con igualdad si y solo si $P(y) = Q(y)$ para todo $y \in \mathcal{Y}$.
(*Hint:* $\log(x) \leq x - 1$).

Propuesta inicial

Busco $\hat{P}(y|x)$ que minimice:

$$\underbrace{\mathbb{E}\left[KL\left(P_{Y|X}(\cdot|X)\|\hat{P}(\cdot|X)\right)\right]}_{\text{Kullback Leibler}} = \underbrace{\mathbb{E}\left[-\log\hat{P}(Y|X)\right]}_{\text{Cross-entropy}} - \underbrace{H(Y|X)}_{\text{Entropia condicional}}$$

Cross-Entropy

Optimalidad para $\ell(x, y) = -\log \hat{P}(y|x)$

$$\mathbb{E}\left[-\log \hat{P}(Y|X)\right] \geq H(Y|X)$$

son igualdad si y solo si $\hat{P}(y|x) = P_{Y|X}(y|x)$ para todo (x, y).

Cross-Entropy

Optimalidad para $\ell(x, y) = -\log \hat{P}(y|x)$

$$\mathbb{E}\left[-\log \hat{P}(Y|X)\right] \geq H(Y|X)$$

son igualdad si y solo si $\hat{P}(y|x) = P_{Y|X}(y|x)$ para todo (x, y).

ERM genera estimadores de máxima verosimilitud

$$\frac{1}{n}\sum_{i=1}^{n} -\log \hat{P}(Y_i|X_i) = -\frac{1}{n}\log\left(\prod_{i=1}^{n} \hat{P}(Y_i|X_i)\right)$$

Cross-Entropy

Optimalidad para $\ell(x, y) = -\log \hat{P}(y|x)$

$$\mathbb{E}\left[-\log \hat{P}(Y|X)\right] \geq H(Y|X)$$

son igualdad si y solo si $\hat{P}(y|x) = P_{Y|X}(y|x)$ para todo (x, y).

ERM genera estimadores de máxima verosimilitud

$$\frac{1}{n}\sum_{i=1}^{n} -\log \hat{P}(Y_i|X_i) = -\frac{1}{n}\log\left(\prod_{i=1}^{n} \hat{P}(Y_i|X_i)\right)$$

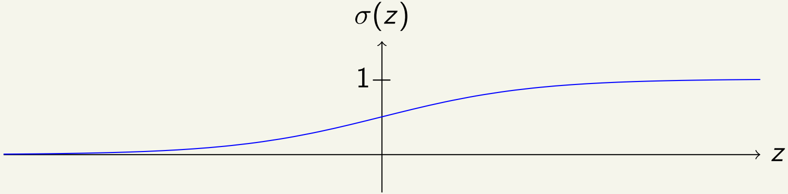
Mismatch de métricas

El mínimo de la cross entropy no tiene por que coincidir exactamente con el mínimo de la probabilidad de error. En general se mira la cross entropy para reducir el bias y la probabilidad de error para prevenir el overfitting.

Regresión Logística Binaria

Función Sigmoide

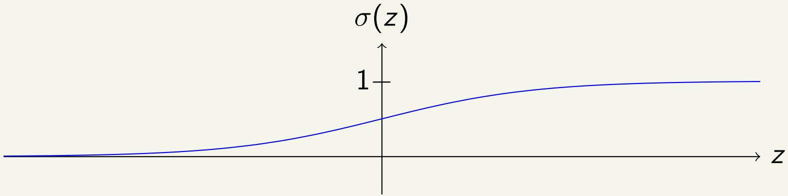
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Regresión Logística Binaria

Función Sigmoide

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



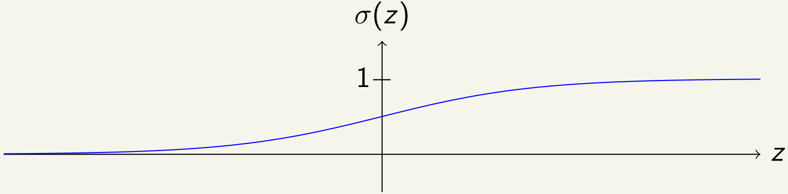
Propuesta

$$\hat{P}(1|x) = \sigma(w^T x + b)$$
$$\hat{P}(0|x) = 1 - \sigma(w^T x + b)$$

Tarea

Función Sigmoide

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



1. Calcular la función inversa $\sigma^{-1}(p)$ con $p \in (0, 1)$.
2. Calcular la derivada $\sigma'(z)$. Encontrar sus valores mínimo y su máximo, y los puntos donde los alcanza.
3. Escribir la derivada en función de $p = \sigma(z)$.

Regresión Logística Binaria

Riesgo empírico

$$\frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i) =$$

$$-\frac{1}{n}\sum_{i=1}^{n} Y_i \log\left(\sigma(w^T X_i + b)\right) + (1 - Y_i) \log\left(1 - \sigma(w^T X_i + b)\right)$$

Regresión Logística Binaria

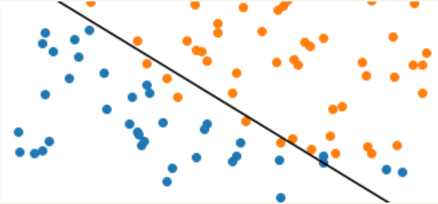
Riesgo empírico

$$\frac{1}{n}\sum_{i=1}^{n}\ell(X_i, Y_i) =$$

$$-\frac{1}{n}\sum_{i=1}^{n} Y_i \log\Big(\sigma(w^T X_i + b)\Big) + (1 - Y_i)\log\Big(1 - \sigma(w^T X_i + b)\Big)$$

Elección del máximo

$$\hat{P}(1|x) \lessgtr \hat{P}(0|x) \quad \Leftrightarrow \quad w^T x + b \lessgtr 0$$



Regresión Logística Categórica (k clases)

Regresión logística clásica

$$\hat{P}(y|x) = \begin{cases} \frac{e^{w_y^T x + b_y}}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y \in \{1, \cdots, k-1\} \\ \\ \frac{1}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y = k \end{cases}$$

Regresión Logística Categórica (k clases)

Regresión logística clásica

$$\hat{P}(y|x) = \begin{cases} \dfrac{e^{w_y^T x + b_y}}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y \in \{1, \cdots, k-1\} \\ \\ \dfrac{1}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y = k \end{cases}$$

Softmax

$$\hat{P}(y|x) = \frac{e^{w_y^T x + b_y}}{\sum_{j=1}^{k} e^{w_j^T x + b_j}}, \qquad y \in \{1, \cdots, k\}$$

Regresión Logística Categórica (k clases)

Regresión logística clásica

$$\hat{P}(y|x) = \begin{cases} \frac{e^{w_y^T x + b_y}}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y \in \{1, \cdots, k-1\} \\ \frac{1}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y = k \end{cases}$$

Softmax

$$\hat{P}(y|x) = \frac{e^{w_y^T x + b_y}}{\sum_{j=1}^{k} e^{w_j^T x + b_j}}, \qquad y \in \{1, \cdots, k\}$$

Riesgo empírico

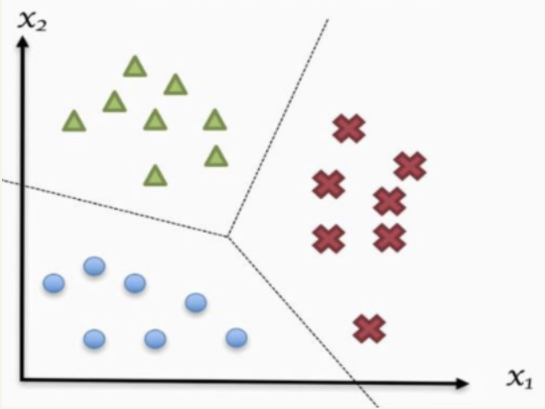
$$\frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i) = \frac{1}{n}\sum_{i=1}^{n} \left[\log\left(\sum_{j=1}^{k} e^{w_j^T X_i + b_j}\right) - \left(w_{Y_i}^T X_i + b_{Y_i}\right)\right]$$

Regresión Softmax

Elección del máximo

$$\arg\max_y \hat{P}(y|x) = \arg\max_y w_y^T x + b_y$$

Se separa con hiper-planos!



Confusion Matrix