Clasificación

Taller de Procesamiento de Señales

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 $\rightarrow \quad \ell(x, y) = \mathbb{1}\{y \neq \varphi(x)\}$
Riesgo Esperado: Probabilidad de error $\rightarrow \quad \mathbb{P}(Y \neq \varphi(X))$

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 $\mathbb{P}(Y \neq \varphi(X))$. 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:

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

Clasificador al azar para k clases

Cualquier clasificador razonable debe ganarle a la decisión al azar:

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

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:

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

Elementos de Teoría de Información

- $H(X) = \mathbb{E}\left[-\log P_X(X)\right]$ Entropía
- $h(X) = \mathbb{E}\left[-\log p_X(X)\right]$ Entropía diferencial
- $H(Y|X) = \mathbb{E}\left[-\log P_{Y|X}(Y|X)\right]$ Entropía condicional
- $h(Y|X) = \mathbb{E}\left[-\log p_{Y|X}(Y|X)\right]$ Entropía diferencial condicional
- $\mathrm{KL}\left(p_X \| q_X\right) = \mathbb{E}\left[\log\left(\frac{p_X(X)}{q_X(X)}\right)\right]$ Divergencia de Kullback Leibler
- $I(X;Y) = \mathrm{KL}\left(p_{XY} \| p_X p_Y\right)$ Información Mutua

Elementos de Teoría de Información

- $H(X) = \mathbb{E}\left[-\log P_X(X)\right]$ Entropía
- $h(X) = \mathbb{E}\left[-\log p_X(X)\right]$ Entropía diferencial
- $H(Y|X) = \mathbb{E}\left[-\log P_{Y|X}(Y|X)\right]$ Entropía condicional
- $h(Y|X) = \mathbb{E}\left[-\log p_{Y|X}(Y|X)\right]$ Entropía diferencial condicional
- $\mathrm{KL}\left(p_X \| q_X\right) = \mathbb{E}\left[\log\left(\frac{p_X(X)}{q_X(X)}\right)\right]$ Divergencia de Kullback Leibler
- $I(X;Y) = \mathrm{KL}\left(p_{XY} \| p_X p_Y\right)$ Información Mutua

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

Propuesta inicial

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

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

Divergencia de Kullback Leibler

Propuesta inicial

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

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

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).

Divergencia de Kullback Leibler

Propuesta inicial

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

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

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).

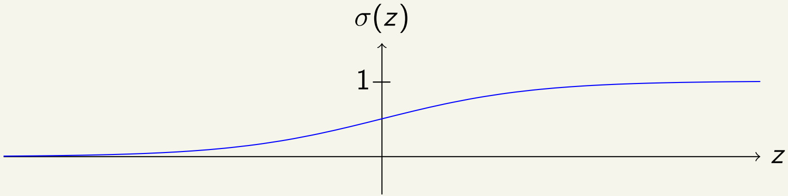
Mismatch de métricas

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

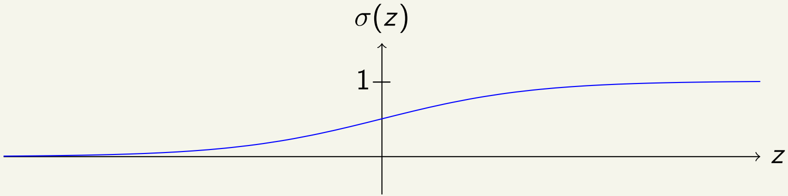
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Regresión Logística Binaria

Función Sigmoide

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



Propuesta

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

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

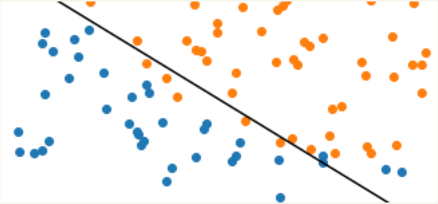
Riesgo empírico

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

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

Elección del máximo

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



Curvas ROC

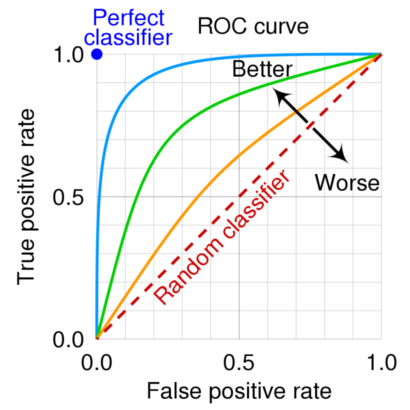
Agregar un umbral

Puedo darle más peso a una clase:

$$w^T x + b \lesseqgtr t$$

$$\text{TPR} = \mathbb{P}(Y = \phi(X)|\text{Es positivo})$$

$$\text{FPR} = \mathbb{P}(Y \neq \phi(X)|\text{Es negativo})$$



Equal Error Rate (EER)

El EER es el error para el cuál los errores tipo I y II coinciden

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

$$\hat{P}(y|x) = \begin{cases} \dfrac{e^{w_y^T x + b_y}}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y \in \{1, \cdots, k-1\} \\[2ex] \dfrac{1}{1+\sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y = k \end{cases}$$

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

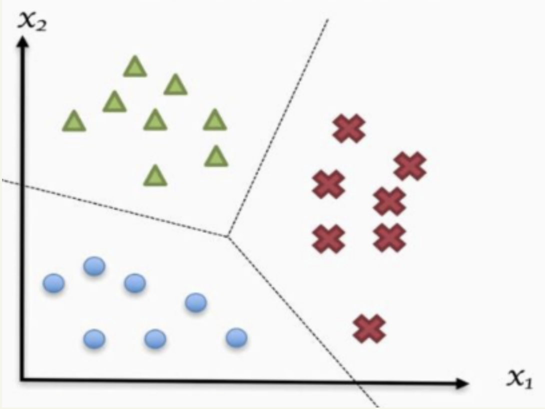
$$\frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i) = \frac{1}{n}\sum_{i=1}^{n} \left[\log\left(\sum_{j=1}^{k} e^{w_j^T X_i + b_j}\right) - \left(w_{Y_i}^T X_i + b_{Y_i}\right)\right]$$

Regresión Softmax

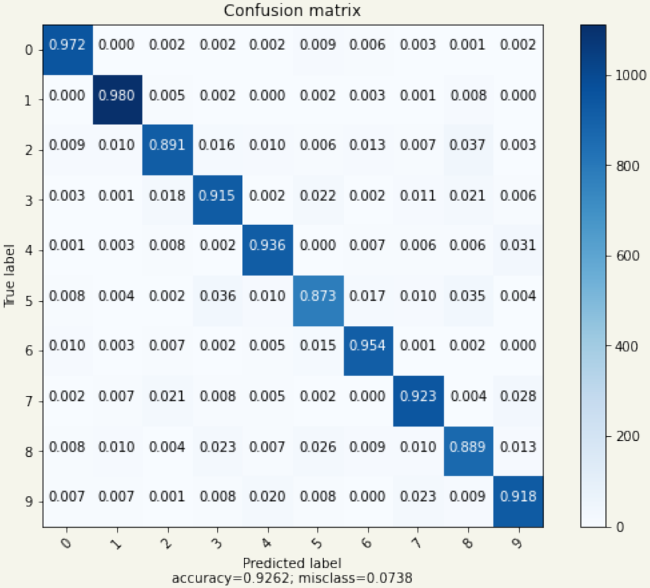
Elección del máximo

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

Se separa con hiperplanos!



Confusion Matrix



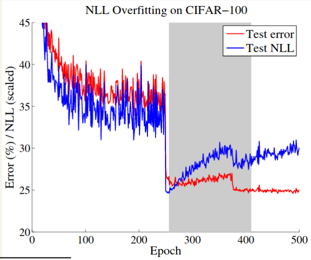
Calibración

Si valido hiperparámetros con respecto a la probabilidad de error, ¿la salida siguen siendo probabilidades?

Para cada par (x, y), existe $0 \leq \zeta \leq \log |\mathcal{Y}|$ tal que

$$-\log \hat{P}(y|x) = -(\mathbf{z}(x))_y + \log\left(\mathbf{1}^T e^{\mathbf{z}(x)}\right) = \max_{y' \in \mathcal{Y}} (\mathbf{z}(x))_{y'} - (\mathbf{z}(x))_y + \zeta$$

Esta concentración de probabilidades es buena para acercarme al clasificador bayesiano, pero puede descalibrar la interpretación probabilística de $\mathbf{f}$.



Guo et al. 2017: "On Calibration of Modern Neural Networks".

Calibración en deep-learning

Temperature Scaling

Se soluciona con la inclusión de un nuevo parámetro (o hiper) $T > 0$:

$$\mathbf{f}(x) = \frac{e^{\mathbf{z}(x)/T}}{\mathbf{1}^T e^{\mathbf{z}(x)/T}}$$

Es importante mirar la cross-entropy en la etapa de validación!

Dataset	Model	Uncalibrated	Temp. Scaling
Birds	ResNet 50	0.9786	**0.8792**
Cars	ResNet 50	0.5488	0.5311
CIFAR-10	ResNet 110	0.3285	0.2102
CIFAR-10	ResNet 110 (SD)	0.2959	0.1718
CIFAR-10	Wide ResNet 32	0.3293	0.2283
CIFAR-10	DenseNet 40	0.2228	**0.1750**
CIFAR-10	LeNet 5	0.4688	0.459
CIFAR-100	ResNet 110	1.4978	**1.0442**
CIFAR-100	ResNet 110 (SD)	1.1157	**0.8613**
CIFAR-100	Wide ResNet 32	1.3434	**1.0565**
CIFAR-100	DenseNet 40	1.0134	0.9026
CIFAR-100	LeNet 5	1.6639	**1.6560**
ImageNet	DenseNet 161	0.9338	0.8885
ImageNet	ResNet 152	0.8961	**0.8657**
SVHN	ResNet 152 (SD)	0.0842	**0.0821**
20 News	DAN 3	0.7949	0.7387
Reuters	DAN 3	0.102	0.0994
SST Binary	TreeLSTM	0.3367	**0.2739**
SST Fine Grained	TreeLSTM	1.1475	1.1168

Guo et al. 2017: "On Calibration of Modern Neural Networks".

Modelos Discriminativos y Generativos

Clasificación de Algoritmos

- **Modelos Discriminativos:** Modelan la dist. condicional $\hat{P}(y|x)$.
- **Modelos Generativos:** Modelan la dist. conjunta $\hat{P}(x, y)$.

Los modelos generativos permiten generar datos sintéticos!

Modelos Discriminativos y Generativos

Clasificación de Algoritmos

- **Modelos Discriminativos:** Modelan la dist. condicional $\hat{P}(y|x)$.
- **Modelos Generativos:** Modelan la dist. conjunta $\hat{P}(x, y)$.

Los modelos generativos permiten generar datos sintéticos!

Linear Discriminant Analysis (LDA)

$$Y \sim \text{Cat}(\{c_1, \cdots, c_K\}), \qquad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma)$$

Linear Discriminant Analysis

Expresiones Matemáticas

$$\hat{p}(x) = \sum_{k=1}^{K} c_k \frac{e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}}{(2\pi)^{d_x/2}|\Sigma|^{1/2}}$$

$$\hat{P}(y|x) = \frac{e^{\mu_y^T \Sigma^{-1} x - \frac{1}{2}\mu_y^T \Sigma^{-1}\mu_y + \log(c_y)}}{\sum_{k=1}^{K} e^{\mu_k^T \Sigma^{-1} x - \frac{1}{2}\mu_k^T \Sigma^{-1}\mu_k + \log(c_k)}}$$

Relación con Regresión Logística

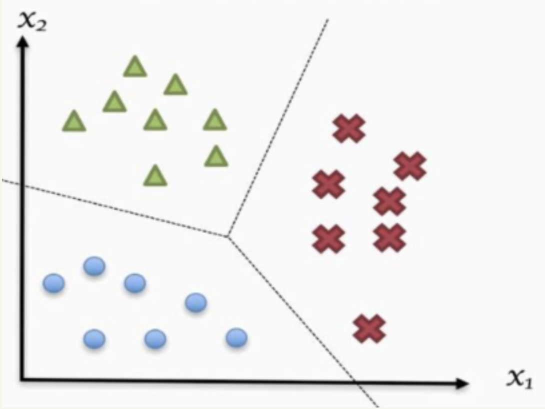
Si $w_y = \Sigma^{-1}\mu_y$ y $b_y = -\frac{1}{2}\mu_y^T \Sigma^{-1}\mu_y + \log(c_y)$, $\hat{P}(y|x)$ es el softmax. LDA utiliza hipótesis más fuertes ya que no solo asume $\hat{P}(y|x)$ softmax, sino también $\hat{p}(x)$ mezcla de gaussianas.

Regresión Softmax

Elección del máximo

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

Se separa con hiperplanos!



Estimación Insesgada de Parámetros

Estimadores

$$\mathcal{D}_k = \{x_i :\ 1 \leq i \leq n \ \wedge\ y_i = k\}$$

$$c_k = \frac{|\mathcal{D}_k|}{n}$$

$$\mu_k = \frac{1}{|\mathcal{D}_k|} \sum_{x \in \mathcal{D}_k} x$$

$$\Sigma_k = \frac{1}{|\mathcal{D}_k| - 1} \sum_{x \in \mathcal{D}_k} (x - \mu_k)(x - \mu_k)^T$$

$$\Sigma = \frac{1}{n - K} \sum_{k=1}^{K} (|\mathcal{D}_k| - 1)\Sigma_k$$

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA)

$$Y \sim \text{Cat}(\{c_1, \cdots, c_K\}), \qquad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA)

$$Y \sim \mathrm{Cat}(\{c_1, \cdots, c_K\}), \qquad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Expresiones Matemáticas

$$\hat{p}(x) = \sum_{k=1}^{K} c_k \frac{e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}}{(2\pi)^{d_x/2}|\Sigma_k|^{1/2}}$$

$$\hat{P}(y|x) = \frac{e^{-\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1}(x-\mu_y)+\log(c_y)-\frac{\log|\Sigma_y|}{2}}}{\sum_{k=1}^{K} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)+\log(c_k)-\frac{\log|\Sigma_k|}{2}}}$$

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA)

$$Y \sim \mathrm{Cat}(\{c_1, \cdots, c_K\}), \qquad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Expresiones Matemáticas

$$\hat{p}(x) = \sum_{k=1}^{K} c_k \frac{e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}}{(2\pi)^{d_x/2}|\Sigma_k|^{1/2}}$$

$$\hat{P}(y|x) = \frac{e^{-\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1}(x-\mu_y)+\log(c_y)-\frac{\log|\Sigma_y|}{2}}}{\sum_{k=1}^{K} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)+\log(c_k)-\frac{\log|\Sigma_k|}{2}}}$$

Elección del máximo

$$\arg\max_{y} \; -\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1}(x-\mu_y) + \log(c_y) - \frac{\log|\Sigma_y|}{2}$$

Modelos Paramétricos y No Paramétricos

Clasificación de Algoritmos

- **Modelos Paramétrcios:** Asumen conocimiento parcial sobre la distribución, indexándola por parámetros.
- **Modelos No Paramétricos:** No se asume una estructura a priori para la distribución.

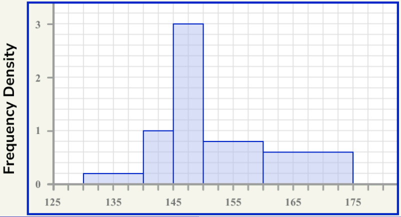
Modelos Paramétricos y No Paramétricos

Clasificación de Algoritmos

- **Modelos Paramétrcios:** Asumen conocimiento parcial sobre la distribución, indexándola por parámetros.
- **Modelos No Paramétricos:** No se asume una estructura a priori para la distribución.

Histograma

El histograma asume una densidad constante por regiones. En cada región asigna $\hat{p}(x) = \frac{K}{n \cdot V}$ donde n es la cantidad de muestras totales, K la cantidad de muestras en dicha región y V el volumen de la región.



K-Vecinos más cercanos (KNN)

Adaptando el concepto a aprendizaje supervisado

Asumiendo que $\hat{P}(y) = \frac{N_y}{n}$ con N_y el número de muestras de la clase y, y que (en cada región) $\hat{p}(x|y) = \frac{K_y}{N_y \cdot V}$ con K_y la cantidad de muestras que caen en la región de la clase y, se obtiene:

$$\hat{P}(y|x) = \frac{\hat{p}(x|y)\hat{P}(y)}{\sum_{i=1}^{K} \hat{p}(x|i)\hat{P}(i)} = \frac{K_y}{K}$$

Es decir, la proporción de muestras de la clase y en la región.

K-Vecinos más cercanos (KNN)

Adaptando el concepto a aprendizaje supervisado

Asumiendo que $\hat{P}(y) = \frac{N_y}{n}$ con N_y el número de muestras de la clase y, y que (en cada región) $\hat{p}(x|y) = \frac{K_y}{N_y \cdot V}$ con K_y la cantidad de muestras que caen en la región de la clase y, se obtiene:

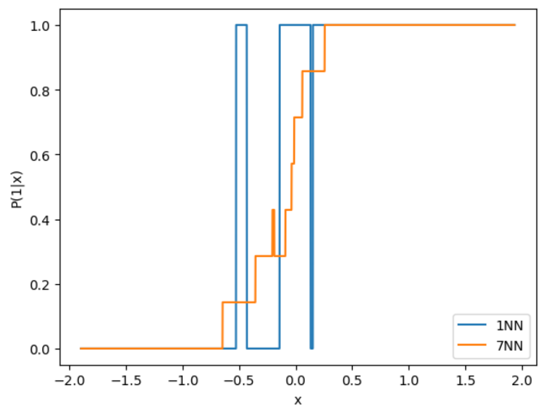
$$\hat{P}(y|x) = \frac{\hat{p}(x|y)\hat{P}(y)}{\sum_{i=1}^{K} \hat{p}(x|i)\hat{P}(i)} = \frac{K_y}{K}$$

Es decir, la proporción de muestras de la clase y en la región.

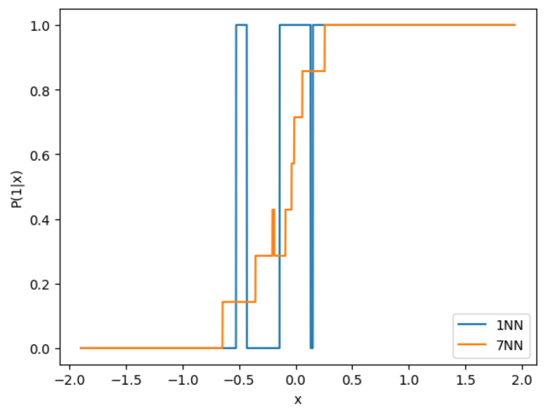
K-Vecinos más cercanos

KNN fija el valor de vecinos K y en base a esto define las regiones. Por ejemplo, la región utilizada para computar un *feature* x es la región centrada en x que posee K muestras (las K más cercanas a x).

K-Vecinos más cercanos (KNN)



K-Vecinos más cercanos (KNN)



Elección del máximo

Notar que para quedarse con el máximo de $\hat{P}(y|x)$ no hace falta computarla. Simplemente se clasifica según sus K vecinos más cercanos, por mayoría.

Support Vector Machines

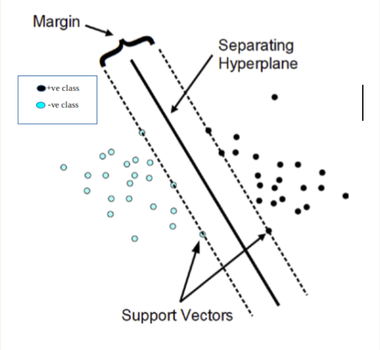
Clases linealmente separables

Sea la clasificación binaria $y \in \{-1, 1\}$ y $z(x) = w^T \cdot x + b$ su frontera de decisión. Decimos que las clases son linealmente separables, si existen w y b tales que $y \cdot z(x) > 0$ para todo $(x, y) \in \mathcal{D}_n$ (set de entrenamiento). Llamamos $f_i(w, b) = y_i z(x_i) > 0$ con $1 \leq i \leq n$.

Support Vector Machines

Clases linealmente separables

Sea la clasificación binaria $y \in \{-1, 1\}$ y $z(x) = w^T \cdot x + b$ su frontera de decisión. Decimos que las clases son linealmente separables, si existen w y b tales que $y \cdot z(x) > 0$ para todo $(x, y) \in \mathcal{D}_n$ (set de entrenamiento). Llamamos $f_i(w, b) = y_i z(x_i) > 0$ con $1 \leq i \leq n$.



- w es ortogonal a la frontera y por lo tanto $w//(x - x_*)$ con x_* la proyección ortogonal de x sobre la frontera.

$$|w^T(x - x_*)| = \|w\|\|x - x_*\|$$

- Dado que x_* está sobre la frontera, $w^T(x - x_*) = z(x)$ y por lo tanto:

$$d(x) = \|x - x_*\| = \frac{|z(x)|}{\|w\|} = \frac{y \cdot z(x)}{\|w\|}$$

Support Vector Machines

Margen

Se define el margen unilateral como criterio de peor caso:

$$m(w,b) = \min_{1 \leq i \leq n} \frac{y_i(w^T \cdot x_i + b)}{\|w\|} = \frac{1}{\|w\|} \min_{1 \leq i \leq n} f_i(w,b) = \frac{f_k(w,b)}{\|w\|}$$

con k un índice óptimo (función de w y b). Por lo tanto, el problema a resolver es maximizar el margen: $\max_{w,b} \; m(w,b)$.

Support Vector Machines

Margen

Se define el margen unilateral como criterio de peor caso:

$$m(w,b) = \min_{1\leq i\leq n} \frac{y_i(w^T \cdot x_i + b)}{\|w\|} = \frac{1}{\|w\|} \min_{1\leq i\leq n} f_i(w,b) = \frac{f_k(w,b)}{\|w\|}$$

con k un índice óptimo (función de w y b). Por lo tanto, el problema a resolver es maximizar el margen: $\max\limits_{w,b} \; m(w,b)$.

Escala

Sea $\alpha \neq 0$, está claro la decisión $z(x) \gtrless 0$ no se ve afectada si reescalamos los parámetros $w \leftarrow \alpha w$ y $b \leftarrow \alpha b$. Esto mismo ocurre con el margen $m(\alpha w, \alpha b) = m(w,b)$. Con lo cuál no se pierde generalidad al asumir $f_k(w,b) = 1$. Luego $m(w,b) = \frac{1}{\|w\|}$ y $f_i(w,b) \geq 1$ para todo $1 \leq i \leq n$.

Las muestras en las que $f_i(w,b) = 1$ se denominan vectores soporte.

Support Vector Machines

Problema de optimización

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i(w^T x_i + b) \geq 1 \quad (\forall\ 1 \leq i \leq n)$$

Support Vector Machines

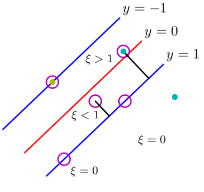
Problema de optimización

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i(w^T x_i + b) \geq 1 \quad (\forall\, 1 \leq i \leq n)$$

Relajando los márgenes

Dado que SVM es un algoritmo de peor caso es muy sensible a outlaiers. Eso hace que muchas veces sea conveniente permitir excepciones a los márgenes. Para ellos se definen las *slack variables*:

$$\xi_i = \begin{cases} 0 & \text{Si} \quad y_i z(x_i) \geq 1 \\ |y_i - z(x_i)| & \text{Si} \quad y_i z(x_i) < 1 \end{cases}$$



La nueva restricción es:

$$y_i z(x_i) \geq 1 - \xi_i$$

Support Vector Machines

Problema de optimización relajando el margen

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{s.t.} \quad \left\{ \begin{array}{c} y_i(w^T x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{array} \right. \qquad (\forall\, 1 \leq i \leq n)$$

Generalización a fronteras no lineales

Este método es fácilmente generalizable a diferentes familias de fronteras $z(x) = w^T\phi(x) + b$, donde la transformación $\phi(\cdot)$ se denomina Kernell.

Árboles de decisión

FALTA

Bosques aleatorios

FALTA