Apprentissage (supervisé) → Régression linéaire

↳ Descente de gradient

→ Résolution des équations des équations normales

→ Régularisation (normes $l_p$)

↳ Ridge $\left( \sum_{j=1}^{D} \beta_j^2 \right)$

→ LASSO $\left( \sum_{j=1}^{D} |\beta_j| \right)$

→ Sélection de caractéristiques

→ Estimateurs Statistiques

→ Maximum de vraissemblance (MLE)

⇔ minimisation somme des carrés des résidus (OLS)

→ Maximum a Posteriori (MAP)

↳ Dans le cas d'un a priori

→ A priori Gaussien ⇒ Ridge

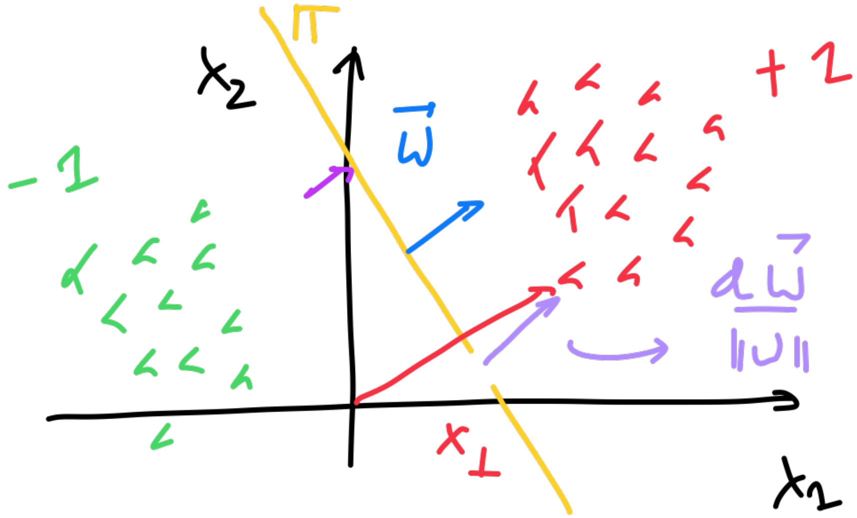
→ A priori Laplace ⇒ LASSO

→ Equilibre Biais Variance

$MSE = biais^2 + variance$

Aujourd'hui → Classification linéaire

$\vec{w} = (\beta_1, \beta_2)$ vecteur normal au plan




$$y(x) = \beta_0 + \underbrace{\beta_1} x_1 + \underbrace{\beta_2} x_2$$

pour tout $x \in \mathbb{R}^2$ on a

$$x = x_\perp + d\frac{\vec{w}}{\|w\|}$$

$$\underbrace{x^\perp \in \pi} \Rightarrow \beta_0 + (\beta_1, \beta_2)^T x_\perp = 0$$

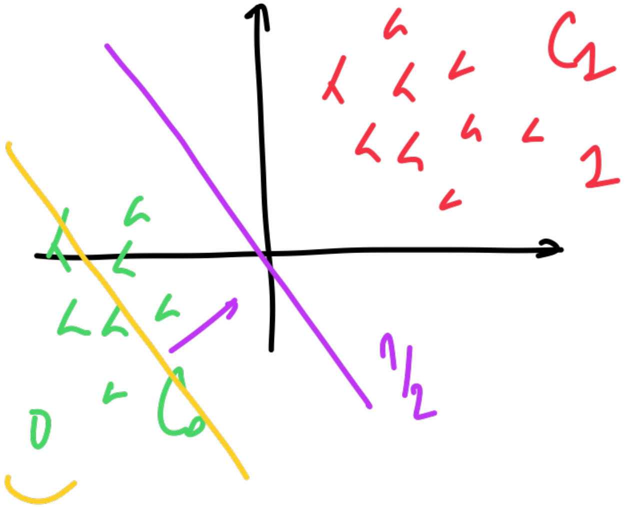
$$\beta_0 + (\beta_1, \beta_2)^T x = \overbrace{\beta_0 + (\beta_1, \beta_2)^T x_\perp}^{=0} + \overbrace{(\beta_1, \beta_2)}^{\vec{w}}{}^T d\frac{\vec{w}}{\|w\|}$$

$$= \begin{cases} d\frac{\|w\|^2}{\|w\|} > 0 & \text{si } x \text{ est situé au dessus de } \pi \\ < 0 & x \text{ est situé en dessous} \end{cases}$$

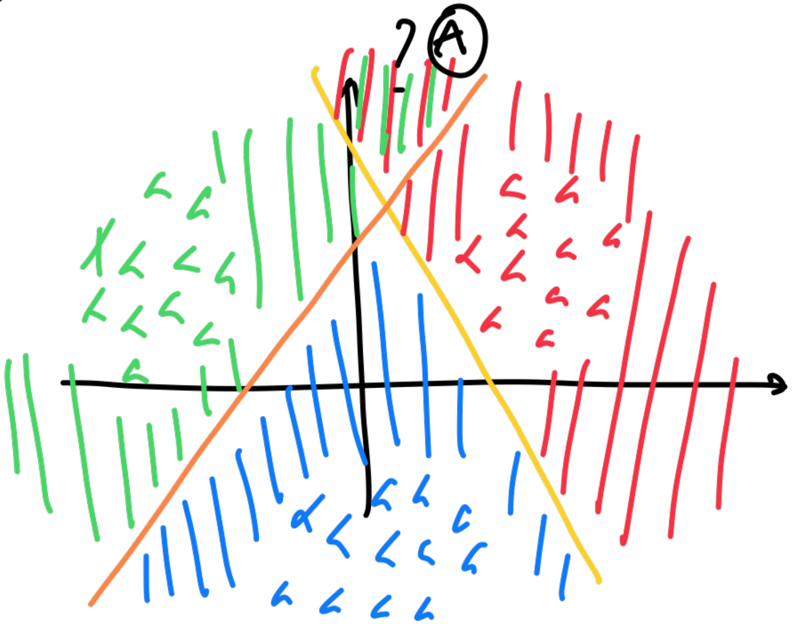
On peut donc utiliser le modèle linéaire pour séparer l'espace en deux classes $C_1$ et $C_2$ correspondant aux 2 demi-espaces (aux deux régions $R_1 = \{ x \in \mathbb{R}^2 \mid y(x) \geq 0 \}$
$R_2 = \{ x \in \mathbb{R}^2 \mid y(x) < 0 \}$

Comme pour la régression linéaire, on peut entraîner le modèle à l'aide de la fonction de coût OLS

$$\ell(\beta) = \frac{1}{N} \sum_{i=1}^{N} \Big( \overset{+1}{\underset{-1}{t^{(i)}}} - \big( \beta_0 + \beta_1 \overset{x_1^{(i)}}{\underset{x_1^{(i)}}{x_1^{(i)}}} + \beta_2 \overset{x_1^{(i)}}{\underset{x_2^{(i)}}{x_2^{(i)}}} \big) \Big)^2$$

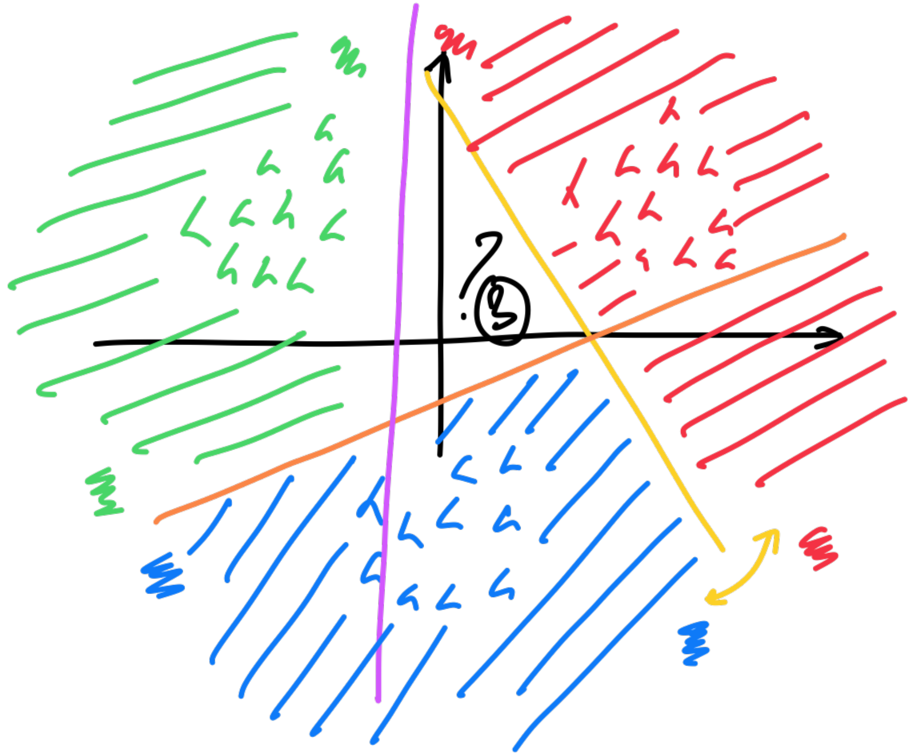



Question ? → Multi-classes ?




pour K classes, le modèle nécessite K-1 discriminants

→ Ambiguïté au niveau de A




Extension #2 : 1 contre 1

→ indétermination (B)

pour K classes : le modèle nécessite

$$\frac{K(K-1)}{2}$$

---

De façon à lever les ambiguïtés on peut considérer K discriminants qu'on entraîne simultanément

pour chaque point $x^{(i)}$ on introduit le vecteur cible $\vec{t}^{(i)}$

$$\vec{t}^{(i)} = [0, 0, 1, 0 \dots 0] \longrightarrow \text{si } x^{(i)} \in C_3$$

On stocke les vecteurs cibles $\vec{t}^{(i)}$ dans la matrice $\underline{\underline{T}}$

$$\underline{\underline{T}} = \begin{bmatrix} — & \vec{t}^{(1)} & — \\ & \vdots & \\ — & \vec{t}^{(N)} & — \end{bmatrix} \in \mathbb{R}^{N \times K} \qquad \underline{\underline{\tilde{X}}} = \begin{bmatrix} 1 & — & \vec{x}^{(1)} & — \\ \vdots & & \vdots & \\ 1 & — & \vec{x}^{(N)} & — \end{bmatrix} \in \mathbb{R}^{N \times D+1}$$

On introduit une matrice de discriminants

$$\underline{\underline{W}} = \begin{bmatrix} | & | & & | \\ \vec{w}^{(1)} & \vec{w}^{(2)} & \dots & \vec{w}^{(K)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{D+1 \times K} \qquad l(\beta) = \sum_i \varepsilon_i^2$$

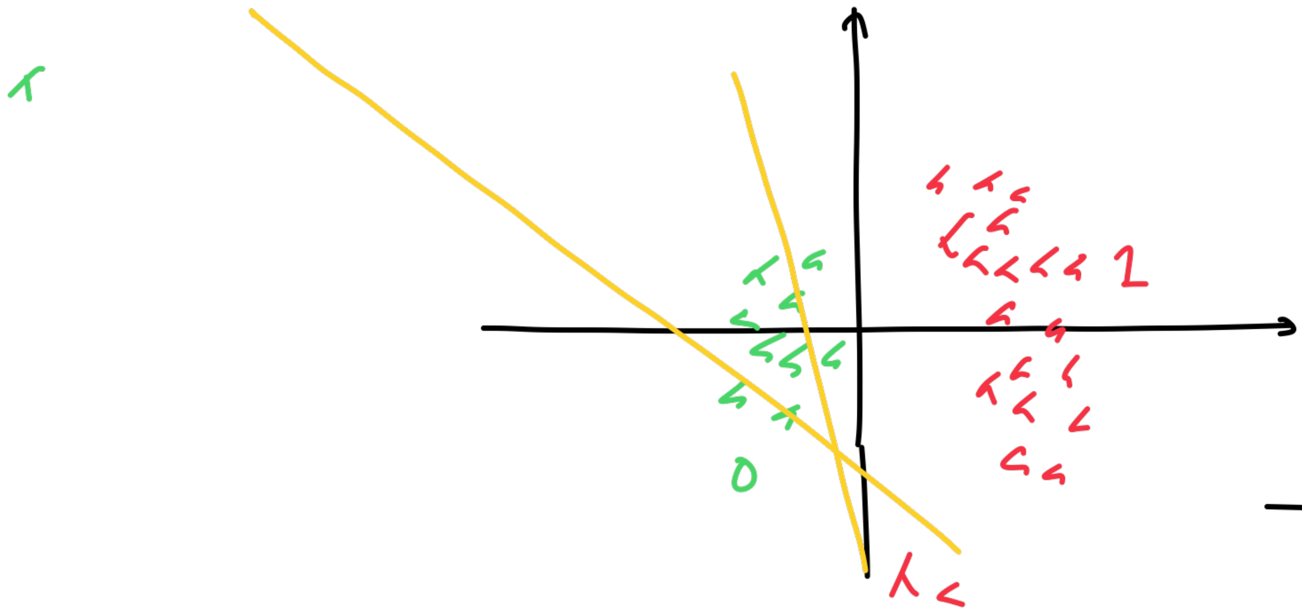
où $\vec{w}^{(k)}$ représente le discriminant associé à la classe $C_k$

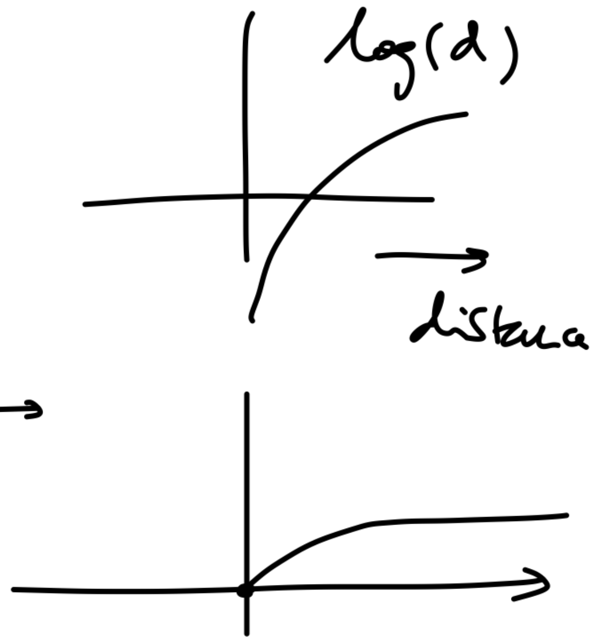
$$E = (T - \underline{\underline{\tilde{X}}}\,\underline{\underline{W}}) \longrightarrow \text{matrice des résidus}$$

On peut définir la fonction de coût

$$\ell(\underline{\underline{W}}) = \frac{1}{KW} \sum_{i=1}^{W} \sum_{j=1}^{K} \left( T - \underline{\underline{\tilde{X}}}\,\underline{\underline{W}} \right)^2_{ij}$$

$\longrightarrow$ la minimisation de $\ell(\underline{\underline{W}})$ donne le modèle à K discriminants qui permet de lever les ambiguïtés des modèles 1 contre 1 et 1 contre tous.

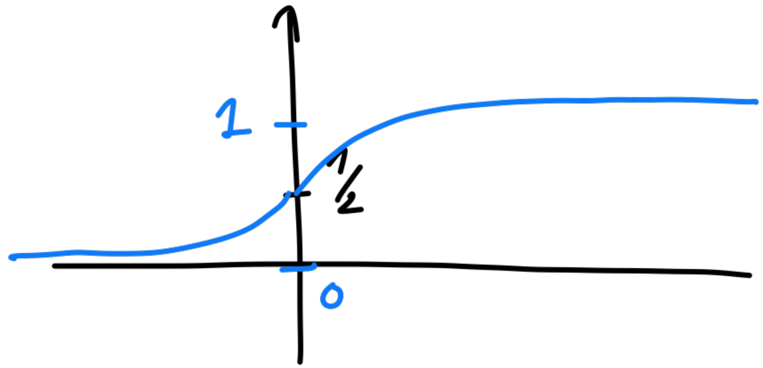


→ Idée : On utilise une fonction d'activation

$\sigma(x)$ et on va définir le modèle comme $\sigma(\beta_0 + \beta_1 x_1 + \beta_1 x_2)$

choix #1 $\qquad \sigma(x) = \dfrac{1}{1 + e^{-x}}$




→ Modèle correspondant est appelé Régression logistique

$$y(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

$$
\left.\begin{aligned}
p(t^{(i)}=1 \mid x^{(i)}, \beta) &= \sigma(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_D x_D^{(i)}) \\
p(t^{(i)}=0 \mid x^{(i)}, \beta) &= 1 - \sigma(\beta_0 + \beta_1 x_2^{(i)} + \dots + \beta_D x_D^{(i)})
\end{aligned}\right\}
$$

$\rightarrow$ Maximum de vraisemblance

$$
\begin{aligned}
p(t^{(i)}=t \mid x^{(i)}, \beta) = {} & \sigma(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_D x_D^{(i)})^{t} \\
& \times (1 - \sigma(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_D x_D^{(i)}))^{1-t}
\end{aligned}
$$

$$
\begin{aligned}
\underbrace{p}_{\text{On suppose l'independance}}(\{t^{(i)}\}_{i=1}^{N} \mid \{x^{(i)}\}_{i=1}^{N}, \vec{\beta}) = \log\Big( & \prod_{i=1}^{N} \sigma(\beta_0 + \beta_1 x_1 + \dots + \beta_D x_D)^{t} \\
& \times (1 - \sigma(\beta_0 + \beta_1 x_1 + \dots + \beta_D x_D))^{1-t} \Big)
\end{aligned}
$$

On considère la fonction de log vraissemblance

$$\ell(\beta) = -\sum_{i=1}^{N} t \log\left(\sigma(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_D x_D^{(i)})\right) + (1-t)\log\left(1-\sigma(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_D x_D^{(i)})\right)$$

Entropie Binaire croisée $\qquad h_\beta = \beta_0 + \beta_1 x_1 + \dots + \beta_D x_D$

$$\rightarrow t\log\left(\sigma(\overbrace{h_\beta(x)}^{z})\right) + (1-t)\log\left(1-\sigma(\overbrace{h_\beta(x)}^{z})\right)$$

$$\sigma'(x) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = \frac{e^{-x}}{(1+e^{-x})^2} = \overbrace{\frac{1}{1+e^{-x}}}^{\sigma(x)}\overbrace{\left(1-\frac{1}{1+e^{-x}}\right)}^{1-\sigma(x)}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

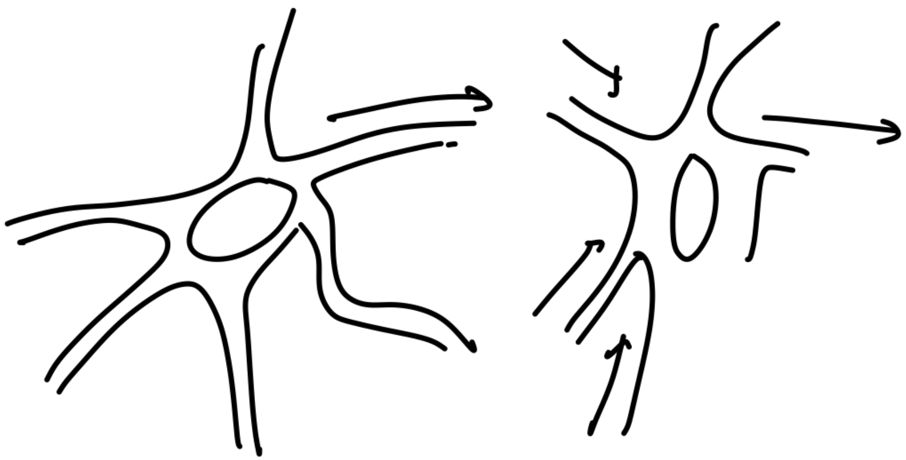
$$\frac{\partial l}{\partial \beta_j} = \frac{\partial l}{\partial h_\beta} \cdot \frac{\partial h_\beta}{\partial \beta_j} = \left( t \, \frac{\sigma'}{\sigma} + (1-t) \, \frac{(-\sigma')}{1-\sigma} \right) \cdot \tilde{x}_j$$

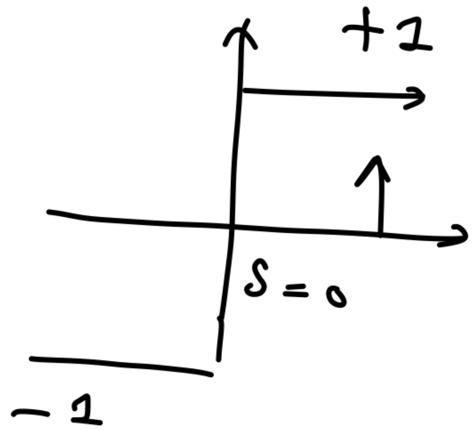
$$= \left( t(1-\sigma) - (1-t) \cdot \sigma \right) \tilde{x}_j$$

$$= \left( t - \sigma(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_D x_D^{(i)}) \right) \cdot \tilde{x}_j^{(i)}$$

---

Option #2





Heaviside $H(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$

$\longrightarrow$ Algorithme du perceptron

$$\min \; - \sum_{i \in \text{misclassified}} \overbrace{t^{(i)}}^{>0} \Big( \underbrace{\beta_0 + \overbrace{\beta_1 x_1^{(i)}}^{<0} + \dots + \beta_D x_D^{(i)}}_{} \Big).$$