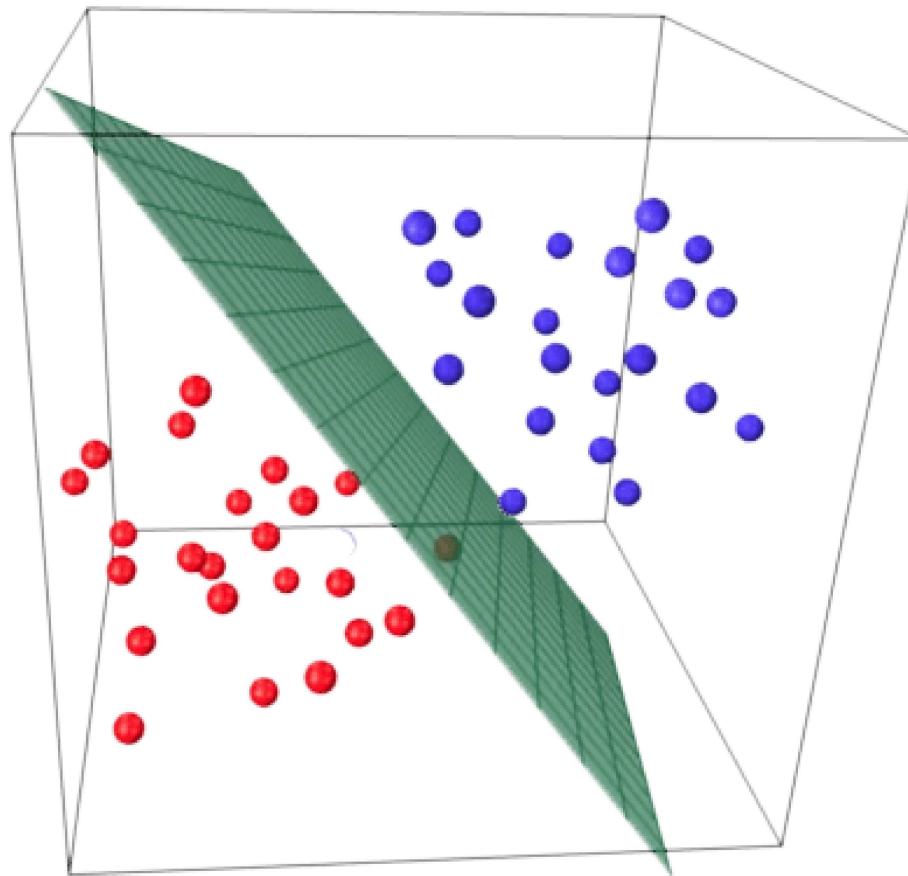


Classificateurs linéaires



Problème de classification binaire

- **Données** $x \in \mathbb{R}^d$ et **Étiquettes** $y \in \{-1, 1\}$
- **Paramètres** : $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Frontière de décision** : $w \cdot x + b = 0$
- **Prédiction** de l'étiquette y d'une donnée x : $y = ?$



Problème de classification binaire

- **Données** $x \in \mathbb{R}^d$ et **Étiquettes** $y \in \{-1, 1\}$
- **Paramètres** : $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Frontière de décision** : $w \cdot x + b = 0$
- **Prédiction** de l'étiquette y d'une donnée x : $y = \text{sign}(w \cdot x + b)$



Problème de classification binaire

- **Données** $x \in \mathbb{R}^d$ et **Étiquettes** $y \in \{-1, 1\}$
- **Paramètres** : $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Frontière de décision** : $w \cdot x + b = 0$
- **Prédiction** de l'étiquette y d'une donnée x : $y = \text{sign}(w \cdot x + b)$
- **Vérification** de l'étiquetage : y est correcte pour x si ? .



Problème de classification binaire

- **Données** $x \in \mathbb{R}^d$ et **Étiquettes** $y \in \{-1, 1\}$
- **Paramètres** : $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Frontière de décision** : $w \cdot x + b = 0$
- **Prédiction** de l'étiquette y d'une donnée x : $y = \text{sign}(w \cdot x + b)$
- **Vérification** de l'étiquetage : y est correcte pour x si $y(w \cdot x + b) > 0$.



Problème de classification binaire

- **Données** $x \in \mathbb{R}^d$ et **Étiquettes** $y \in \{-1, 1\}$
- **Paramètres** : $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Frontière de décision** : $w \cdot x + b = 0$
- **Prédiction** de l'étiquette y d'une donnée x : $y = \text{sign}(w \cdot x + b)$
- **Vérification** de l'étiquetage : y est correcte pour x si $y(w \cdot x + b) > 0$.

Fonction objectif à minimiser

$$f(w, b) := \frac{1}{n} \sum_{i=1}^n \max(0, -y^{(i)} (w \cdot x^{(i)} + b))$$



Problème de classification binaire

- **Données** $x \in \mathbb{R}^d$ et **Étiquettes** $y \in \{-1, 1\}$
- **Paramètres** : $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Frontière de décision** : $w \cdot x + b = 0$
- **Prédiction** de l'étiquette y d'une donnée x : $y = \text{sign}(w \cdot x + b)$
- **Vérification** de l'étiquetage : y est correcte pour x si $y(w \cdot x + b) > 0$.

Fonction objectif à minimiser

$$f(w, b) := \frac{1}{n} \sum_{i=1}^n \max(0, -y^{(i)} (w \cdot x^{(i)} + b))$$

Algorithme : le Perceptron

- **Initialisation** $w = 0$ et $b = 0$
- **Itérations** Boucle sur les données (x, y) (jusqu'à stationnarité) :
 - Si $y(w \cdot x + b) \leq 0$: $w \leftarrow ?$ et $b \leftarrow ?$



Problème de classification binaire

- **Données** $x \in \mathbb{R}^d$ et **Étiquettes** $y \in \{-1, 1\}$
- **Paramètres** : $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Frontière de décision** : $w \cdot x + b = 0$
- **Prédiction** de l'étiquette y d'une donnée x : $y = \text{sign}(w \cdot x + b)$
- **Vérification** de l'étiquetage : y est correcte pour x si $y(w \cdot x + b) > 0$.

Fonction objectif à minimiser

$$f(w, b) := \frac{1}{n} \sum_{i=1}^n \max(0, -y^{(i)}(w \cdot x^{(i)} + b))$$

Algorithme : le Perceptron

- **Initialisation** $w = 0$ et $b = 0$
- **Itérations** Boucle sur les données (x, y) (jusqu'à stationnarité) :
 - Si $y(w \cdot x + b) \leq 0$: $w \leftarrow w + yx$ et $b \leftarrow b + y$



Remarque

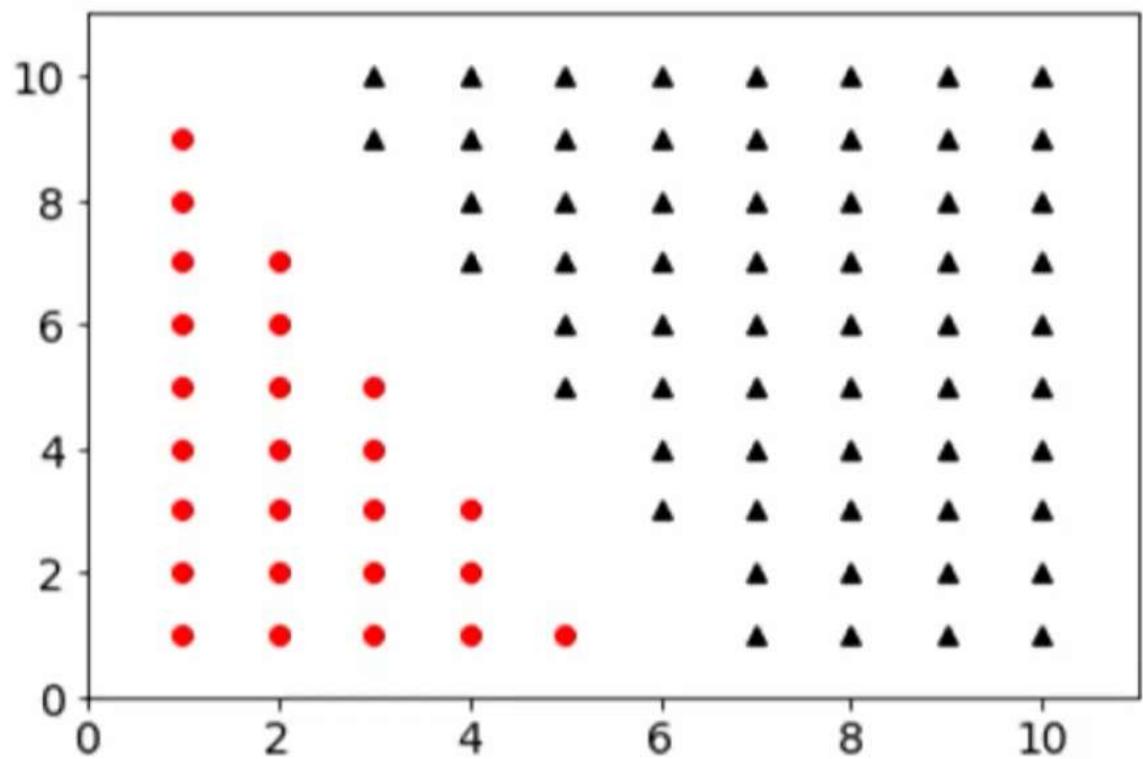
*L'algorithme du Perceptron consiste donc à minimiser la fonction f pour déterminer w et b en utilisant une **descente de gradient stochastique**.*



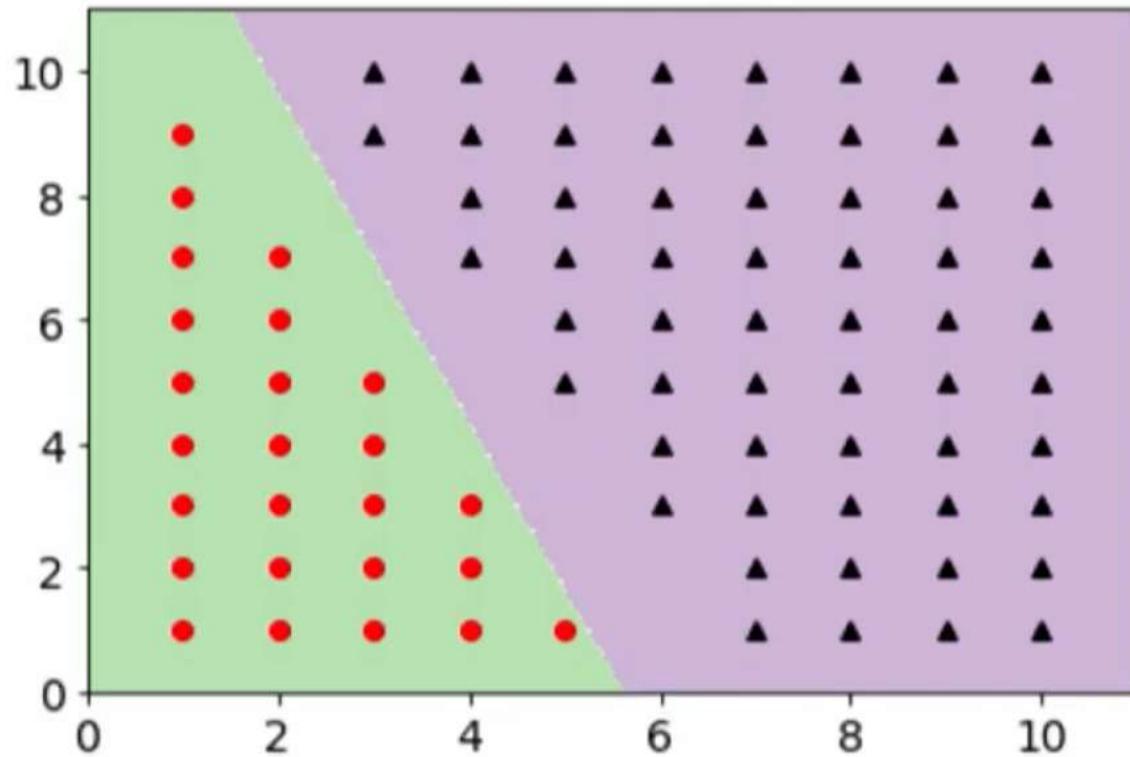
Remarque

*L'algorithme du Perceptron consiste donc à minimiser la fonction f pour déterminer w et b en utilisant une **descente de gradient stochastique**. Elle diffère de la méthode de gradient classique dans le sens où à la i ème itération de l'algorithme du Perceptron, la direction de descente est donnée par le gradient du i ème terme de la fonction objectif (au lieu de prendre le gradient total de la fonction objectif).*





8 iterations



Convergence

Si les données sont **linéairement séparables** :

- Il existe (au moins) une frontière de décision.

Q. Que vaut le coût lorsque l'algorithme trouve cette frontière ?



Convergence

Si les données sont **linéairement séparables** :

- Il existe (au moins) une frontière de décision.
R. L'algorithme va trouver une frontière de décision avec un **coût nul**



Convergence

Si les données sont **linéairement séparables** :

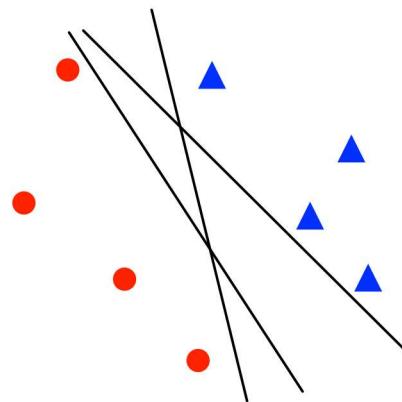
- Il existe (au moins) une frontière de décision.
R. L'algorithme va trouver une frontière de décision avec un **coût nul**
- L'algorithme converge en un nombre fini d'itérations



Convergence

Si les données sont **linéairement séparables** :

- Il existe (au moins) une frontière de décision.
R. L'algorithme va trouver une frontière de décision avec un **coût nul**
- L'algorithme converge en un nombre fini d'itérations



Q. Comment calculer la "meilleure" frontière de décision ?

Remarque sur le dual du Perceptron

Algorithme du Perceptron (primal)

- **Initialisation** $w = 0$ et $b = 0$
- **Itérations** Boucle sur les données $(x^{(i)}, y^{(i)})$:
 - Si $y^{(i)} (w \cdot x^{(i)} + b) \leq 0$: $w \leftarrow w + y^{(i)} x^{(i)}$ et $b \leftarrow b + y^{(i)}$

Algorithme du Perceptron (dual)

- **Initialisation** $\lambda = 0$ et $b = 0$
- **Itérations** Boucle sur les données $(x^{(i)}, y^{(i)})$:
 - Si $(x^{(i)}, y^{(i)})$ n'est pas bien classifié : $\lambda_i \leftarrow \lambda_i + 1$ et $b \leftarrow b + y^{(i)}$



Remarque sur le dual du Perceptron

Algorithme du Perceptron (primal)

- **Initialisation** $w = 0$ et $b = 0$
- **Itérations** Boucle sur les données $(x^{(i)}, y^{(i)})$:
 - Si $y^{(i)} (w \cdot x^{(i)} + b) \leq 0$: $w \leftarrow w + y^{(i)} x^{(i)}$ et $b \leftarrow b + y^{(i)}$

Algorithme du Perceptron (dual)

- **Initialisation** $\lambda = 0$ et $b = 0$
- **Itérations** Boucle sur les données $(x^{(i)}, y^{(i)})$:
 - Si $(x^{(i)}, y^{(i)})$ n'est pas bien classifié : $\lambda_i \leftarrow \lambda_i + 1$ et $b \leftarrow b + y^{(i)}$

\Rightarrow cette version duale est plus simple car à chaque itération, on manipule deux scalaires (et non des vecteurs comme dans le primal).

λ_i = nombre de mise à jour du point $(x^{(i)}, y^{(i)})$.



Remarque sur le dual du Perceptron

Algorithme du Perceptron (primal)

- **Initialisation** $w = 0$ et $b = 0$
- **Itérations** Boucle sur les données $(x^{(i)}, y^{(i)})$:
 - Si $y^{(i)}(w \cdot x^{(i)} + b) \leq 0$: $w \leftarrow w + y^{(i)}x^{(i)}$ et $b \leftarrow b + y^{(i)}$

Algorithme du Perceptron (dual)

- **Initialisation** $\lambda = 0$ et $b = 0$
- **Itérations** Boucle sur les données $(x^{(i)}, y^{(i)})$:
 - Si $(x^{(i)}, y^{(i)})$ n'est pas bien classifié : $\lambda_i \leftarrow \lambda_i + 1$ et $b \leftarrow b + y^{(i)}$

\Rightarrow cette version duale est plus simple car à chaque itération, on manipule deux scalaires (et non des vecteurs comme dans le primal).

λ_i = nombre de mise à jour du point $(x^{(i)}, y^{(i)})$.

On verra que w peut être reconstruit avec

$$w = \sum_{i=1}^n \lambda_i y^{(i)} x^{(i)}.$$



Amélioration du Perceptron : **SVM**

Données d'entraînement : $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, 1\}$

But : Trouver $w \in \mathbb{R}^d$ et $b \in \mathbb{R}$ tels que $y^{(i)}(w \cdot x^{(i)} + b) > 0$ pour tout $i = 1, \dots, n$.
En renormalisant w et b , cette condition peut s'écrire

$$y^{(i)}(w \cdot x^{(i)} + b) \geq 1, \quad \forall i = 1, \dots, n,$$

ce qui nous permet d'avoir une formule simple pour la marge.



Amélioration du Perceptron : SVM

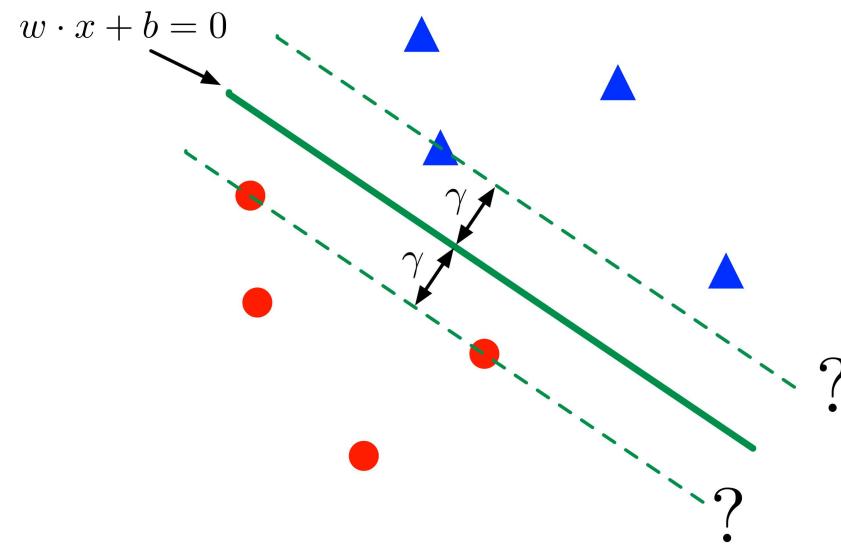
Données d'entraînement : $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, 1\}$

But : Trouver $w \in \mathbb{R}^d$ et $b \in \mathbb{R}$ tels que $y^{(i)}(w \cdot x^{(i)} + b) > 0$ pour tout $i = 1, \dots, n$.
En renormalisant w et b , cette condition peut s'écrire

$$y^{(i)}(w \cdot x^{(i)} + b) \geq 1, \quad \forall i = 1, \dots, n,$$

ce qui nous permet d'avoir une formule simple pour la marge.

Maximiser la marge γ



Amélioration du Perceptron : SVM

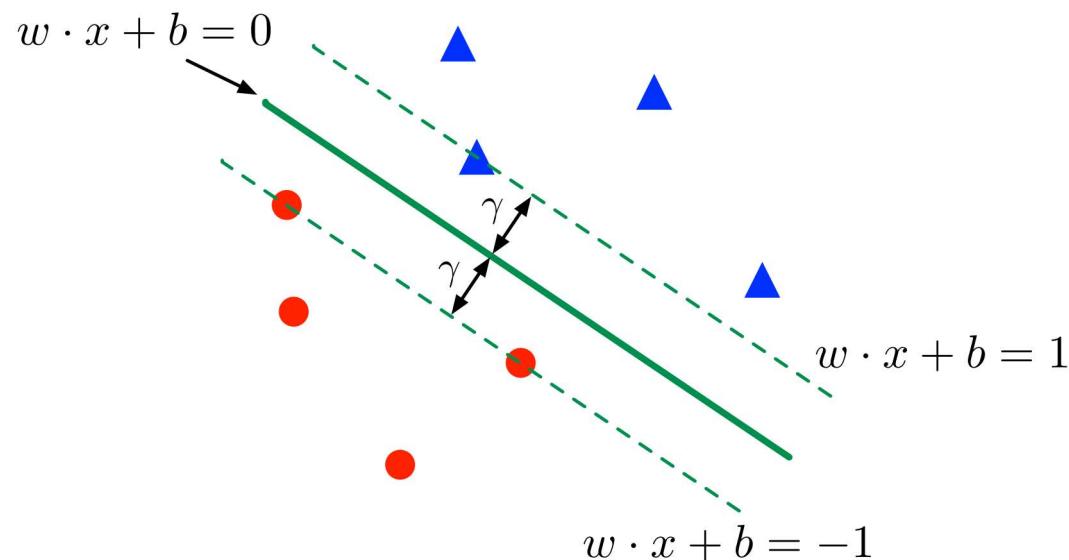
Données d'entraînement : $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \{-1, 1\}$

But : Trouver $w \in \mathbb{R}^d$ et $b \in \mathbb{R}$ tels que $y^{(i)}(w \cdot x^{(i)} + b) > 0$ pour tout $i = 1, \dots, n$.
En renormalisant w et b , cette condition peut s'écrire

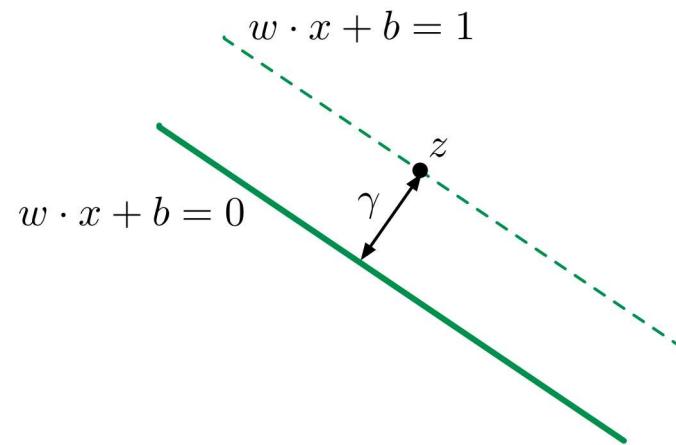
$$y^{(i)}(w \cdot x^{(i)} + b) \geq 1, \quad \forall i = 1, \dots, n,$$

ce qui nous permet d'avoir une formule simple pour la marge.

Maximiser la marge γ



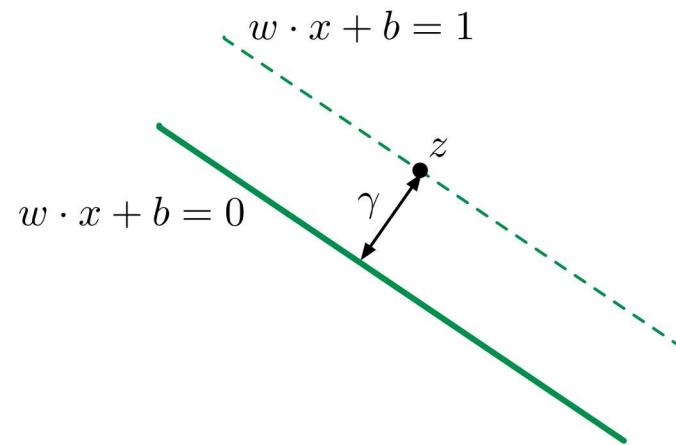
Formule de la marge. Définir γ à partir du point z (correctement classé) le plus proche de la frontière $w \cdot x + b = 0$.



$$\gamma = ?$$



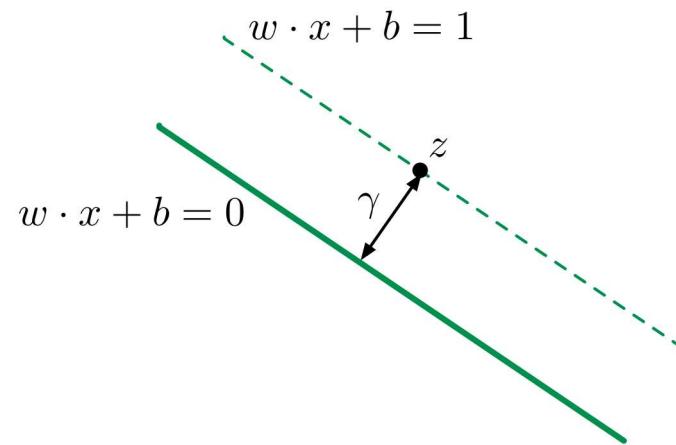
Formule de la marge. Définir γ à partir du point z (correctement classé) le plus proche de la frontière $w \cdot x + b = 0$.



$$\gamma = \frac{1}{\|w\|}$$



Formule de la marge. Définir γ à partir du point z (correctement classé) le plus proche de la frontière $w \cdot x + b = 0$.



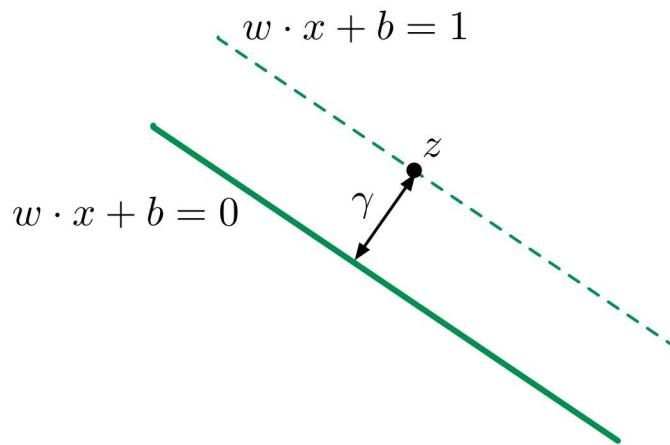
$$\gamma = \frac{1}{\|w\|}$$

Calculs de w, b optimaux (uniques).

$$\max_{w,b} \gamma \Leftrightarrow \min_{w,b} \|w\|$$



Formule de la marge. Définir γ à partir du point z (correctement classé) le plus proche de la frontière $w \cdot x + b = 0$.



$$\gamma = \frac{1}{\|w\|}$$

Calculs de w, b optimaux (uniques).

$$\max_{w,b} \gamma \Leftrightarrow \min_{w,b} \|w\|$$

Choix sur la norme $\|\cdot\|$ à considérer.



Classifieur linéaire qui maximise la marge (SVM).

- Étant donné $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^n \times \{-1, 1\}$, on résout



Classifieur linéaire qui maximise la marge (SVM).

- Étant donné $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^n \times \{-1, 1\}$, on résout

$$\begin{array}{ll} \text{(Primal)} & \min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}} \quad \frac{1}{2} \|\boldsymbol{w}\|^2 \\ & y^{(i)} (\boldsymbol{w} \cdot \boldsymbol{x}^{(i)} + b) \geq 1 \text{ pour tout } i = 1, \dots, n \end{array}$$



Classifieur linéaire qui maximise la marge (SVM).

- Étant donné $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^n \times \{-1, 1\}$, on résout

$$\begin{array}{ll} \text{(Primal)} & \min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad \frac{1}{2} \|w\|^2 \\ & y^{(i)} (w \cdot x^{(i)} + b) \geq 1 \text{ pour tout } i = 1, \dots, n \end{array}$$

- (Primal) est un problème d'optimisation **convexe** en dimension finie, il peut être résolu efficacement



Classifieur linéaire qui maximise la marge (SVM).

- Étant donné $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^n \times \{-1, 1\}$, on résout

$$\begin{array}{ll} \text{(Primal)} & \min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}} \quad \frac{1}{2} \|\boldsymbol{w}\|^2 \\ & y^{(i)} (\boldsymbol{w} \cdot \boldsymbol{x}^{(i)} + b) \geq 1 \text{ pour tout } i = 1, \dots, n \end{array}$$

- (Primal) est un problème d'optimisation **convexe** en dimension finie, il peut être résolu efficacement
- On peut obtenir des informations sur la solution optimale avec la **dualité**



Classifieur linéaire qui maximise la marge (SVM).

- Étant donné $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^n \times \{-1, 1\}$, on résout

$$\begin{array}{ll} \text{(Primal)} & \min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad \frac{1}{2} \|w\|^2 \\ & y^{(i)} (w \cdot x^{(i)} + b) \geq 1 \text{ pour tout } i = 1, \dots, n \end{array}$$

- (Primal) est un problème d'optimisation **convexe** en dimension finie, il peut être résolu efficacement
- On peut obtenir des informations sur la solution optimale avec la **dualité**

Définissons le Lagrangien

$$L(w, b, \lambda) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \lambda_i \left(y^{(i)} (w \cdot x^{(i)} + b) - 1 \right)$$



Dualité

Problème dual

$$\sup_{\lambda \succeq 0} \inf_w L(w, b, \lambda).$$

On a toujours la *dualité faible* :

$$d^* := \sup_{\lambda \succeq 0} \inf_{w,b} L(w, b, \lambda) \leq \inf_{w,b} \sup_{\lambda \succeq 0} L(w, b, \lambda) := p^*.$$

Proposition (Dualité forte)

Si le problème (Primal) est convexe et il existe une solution strictement admissible, alors

$$d^* = p^*.$$



Application Le dual du problème (Primal) s'écrit



Application Le dual du problème (Primal) s'écrit

$$\begin{aligned} & \max_{\lambda \in \mathbb{R}^n} \quad \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{1 \leq i, j \leq n} \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)} \cdot x^{(j)} \\ (\text{Dual}) \quad & \sum_{i=1}^n \lambda_i y^{(i)} = 0 \\ & \lambda_i \geq 0, \quad \forall i = 1, \dots, n \end{aligned}$$



Application Le dual du problème (Primal) s'écrit

$$\begin{aligned}
 & \max_{\lambda \in \mathbb{R}^n} \quad \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{1 \leq i, j \leq n} \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)} \cdot x^{(j)} \\
 (\text{Dual}) \quad & \sum_{i=1}^n \lambda_i y^{(i)} = 0 \\
 & \lambda_i \geq 0, \quad \forall i = 1, \dots, n
 \end{aligned}$$

Le problème (Primal) étant convexe et ayant un point strictement admissible, la **dualité forte** implique que le gap de dualité est nul et les conditions de complémentarité sont

$$\begin{aligned}
 w &= \sum_{i=1}^n \lambda_i y^{(i)} x^{(i)}, \\
 \lambda_i > 0 \Rightarrow y^{(i)} (w \cdot x^{(i)} + b) &= 1.
 \end{aligned}$$



Application Le dual du problème (Primal) s'écrit

$$\begin{aligned} & \max_{\lambda \in \mathbb{R}^n} \quad \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{1 \leq i,j \leq n} \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)} \cdot x^{(j)} \\ (\text{Dual}) \quad & \sum_{i=1}^n \lambda_i y^{(i)} = 0 \\ & \lambda_i \geq 0, \quad \forall i = 1, \dots, n \end{aligned}$$

Le problème (Primal) étant convexe et ayant un point strictement admissible, la **dualité forte** implique que le gap de dualité est nul et les conditions de complémentarité sont

$$\begin{aligned} w &= \sum_{i=1}^n \lambda_i y^{(i)} x^{(i)}, \\ \lambda_i > 0 \Rightarrow y^{(i)} (w \cdot x^{(i)} + b) &= 1. \end{aligned}$$

Les points $x^{(i)}$ pour lesquels $\lambda_i > 0$ sont les **vecteurs de supports**.



Vecteurs de supports Ils correspondent aux données d'entraînements bien classés vérifiant $y^{(i)} (w \cdot x^{(i)} + b) = 1$.

Par dualité, on a vu que w s'écrit :

$$w = \sum_{i=1}^n \lambda_i y^{(i)} x^{(i)}$$

qui s'expriment simplement en fonction des **vecteurs supports**.



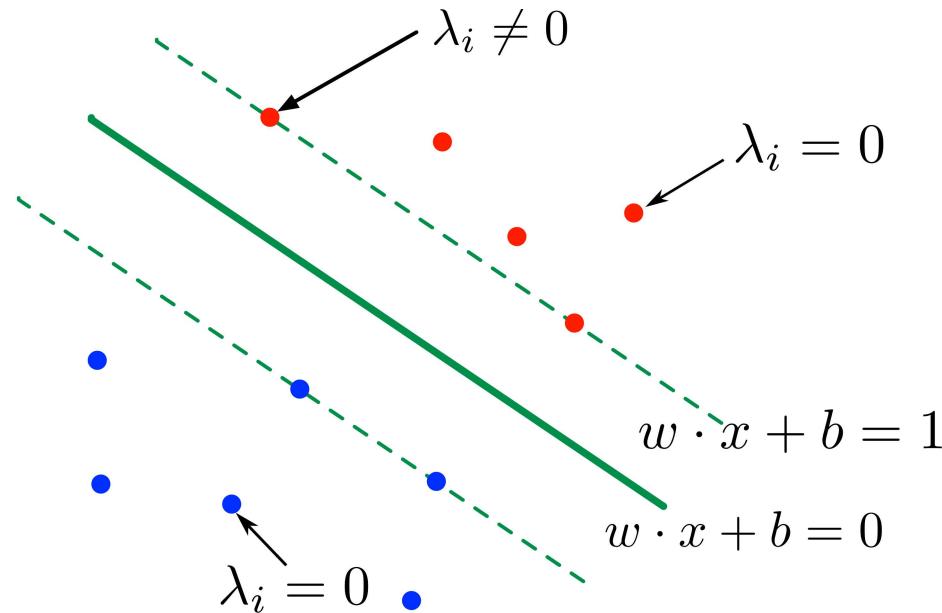
Vecteurs de supports Ils correspondent aux données d'entraînements bien classés vérifiant $y^{(i)} (w \cdot x^{(i)} + b) = 1$.

Par dualité, on a vu que w s'écrit :

$$w = \sum_{i=1}^n \lambda_i y^{(i)} x^{(i)}$$

qui s'exprime simplement en fonction des **vecteurs supports**.

Exercice : Montrer que $\lambda_i = 0$ pour tout i tel que $y^{(i)} (w \cdot x^{(i)} + b) \neq 1$.

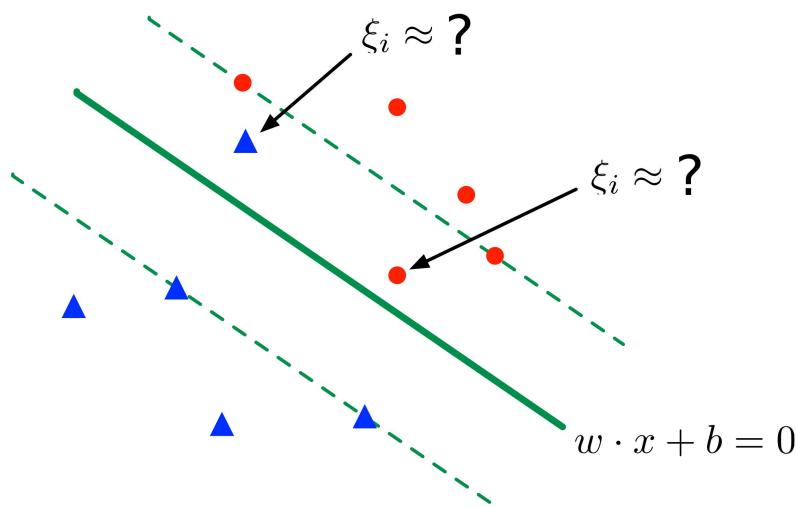


Le cas linéairement non-séparable

L'erreur d'entraînement est non nulle.

Méthode Ajout d'une variable "d'écart" ("slack") $\xi \in \mathbb{R}^n$ pour **relâcher les contraintes** et ξ_i est associée à chaque inégalité du problème (Primal), on obtient le problème :

$$\begin{aligned} & \min_{\mathbf{w} \in \mathbb{R}^d, \mathbf{b} \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^n} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{(Primal')} \quad y^{(i)} (\mathbf{w} \cdot \mathbf{x}^{(i)} + \mathbf{b}) \geq 1 - \xi_i \text{ pour tout } i = 1, \dots, n \\ & \quad \xi_i \geq 0 \text{ pour tout } i = 1, \dots, n. \end{aligned}$$

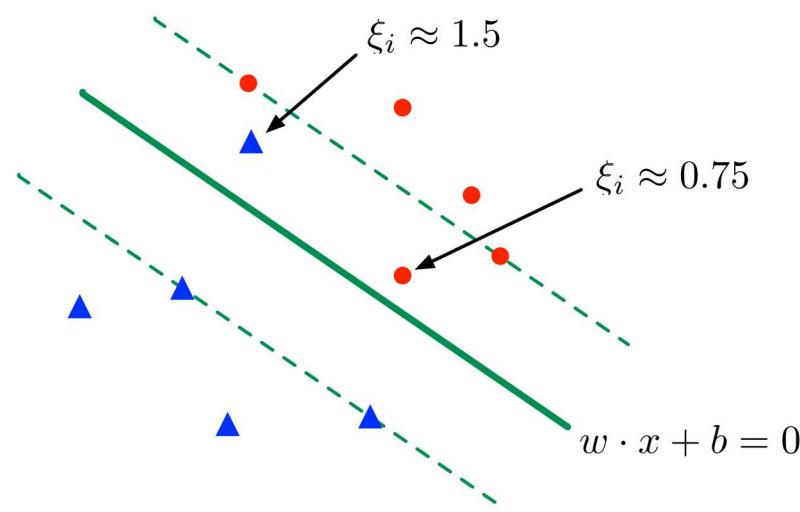


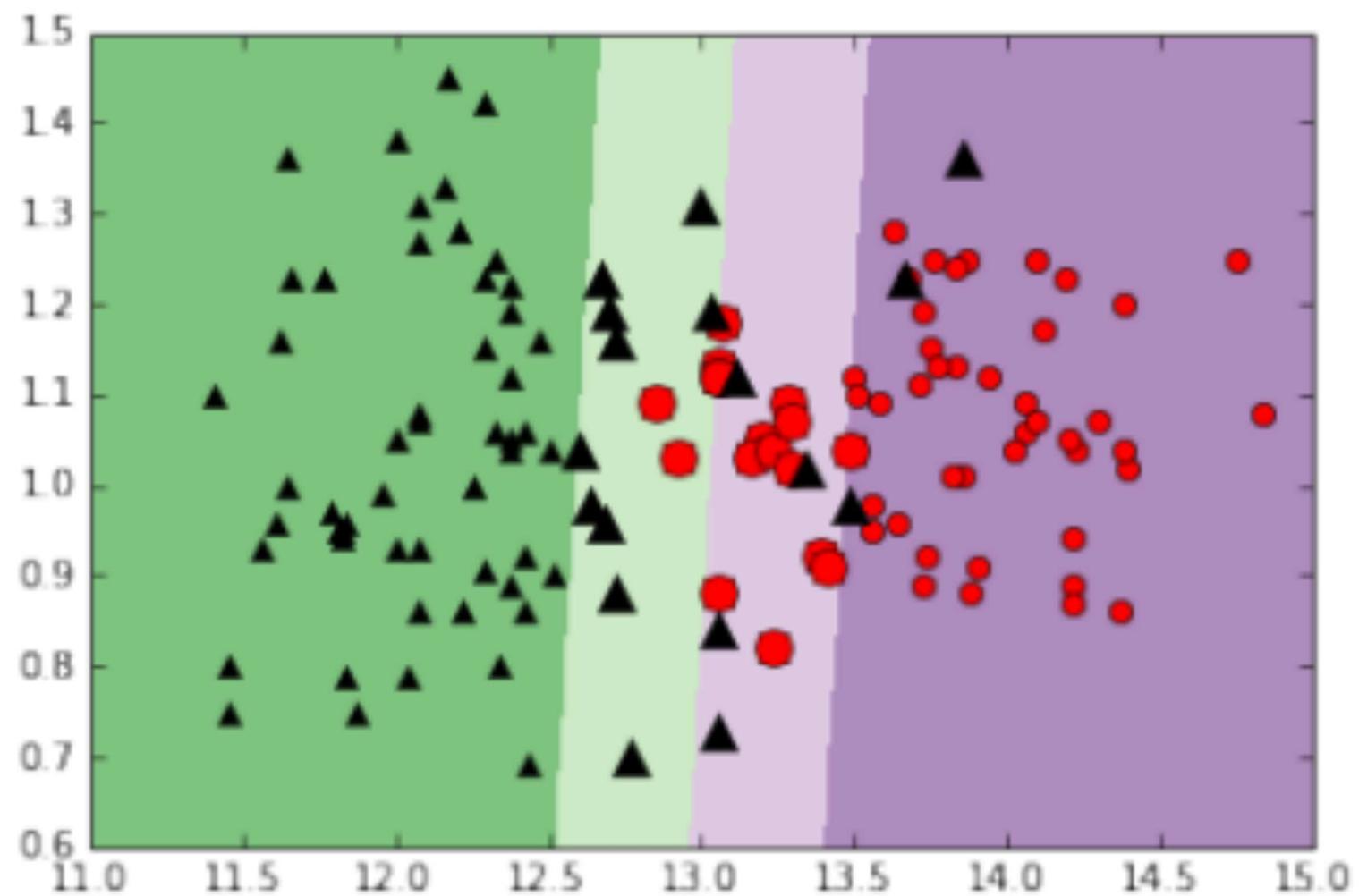
Le cas linéairement non-séparable

L'erreur d'entraînement est non nulle.

Méthode Ajout d'une variable "d'écart" ("slack") $\xi \in \mathbb{R}^n$ pour **relâcher les contraintes** et ξ_i est associée à chaque inégalité du problème (Primal), on obtient le problème :

$$\begin{aligned} & \min_{\mathbf{w} \in \mathbb{R}^d, \mathbf{b} \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ (\text{Primal}') \quad & y^{(i)} (\mathbf{w} \cdot \mathbf{x}^{(i)} + \mathbf{b}) \geq 1 - \xi_i \text{ pour tout } i = 1, \dots, n \\ & \xi_i \geq 0 \text{ pour tout } i = 1, \dots, n. \end{aligned}$$





Le dual du problème (Primal') s'écrit



Le dual du problème (Primal') s'écrit

$$\begin{aligned}
 & \min_{\lambda \in \mathbb{R}^n} \quad \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{1 \leq i, j \leq n} \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)} \cdot x^{(j)} \\
 (\text{Dual}') \quad & \sum_{i=1}^n \lambda_i y^{(i)} = 0 \\
 & 0 \leq \lambda_i \leq C, \quad \forall i = 1, \dots, 2n.
 \end{aligned}$$

La **dualité forte** (convexité et strict faisabilité de (Primal')) implique que le gap de dualité est nul et les conditions de complémentarité sont

$$\begin{aligned}
 w &= \sum_{i=1}^n \lambda_i y^{(i)} x^{(i)}, \\
 0 < \lambda_i < C &\Rightarrow y^{(i)} (\textcolor{blue}{w} \cdot \textcolor{green}{x}^{(i)} + b) = 1, \\
 \lambda_i = C &\Rightarrow y^{(i)} (\textcolor{blue}{w} \cdot \textcolor{green}{x}^{(i)} + \textcolor{blue}{b}) = 1 - \xi_i.
 \end{aligned}$$

On a alors deux types de vecteurs de support dans ce cas.



Comment déterminer la constante C dans la fonction objectif?

C : Trade-off entre la marge et les variables “slacks”.

$C = 0$: les erreurs ne sont pas pénalisées

$C = \infty$: aucune erreur n'est autorisée

Lorsque $C \rightarrow 0$, on obtient une marge plus grande.



Comment déterminer la constante C dans la fonction objectif?

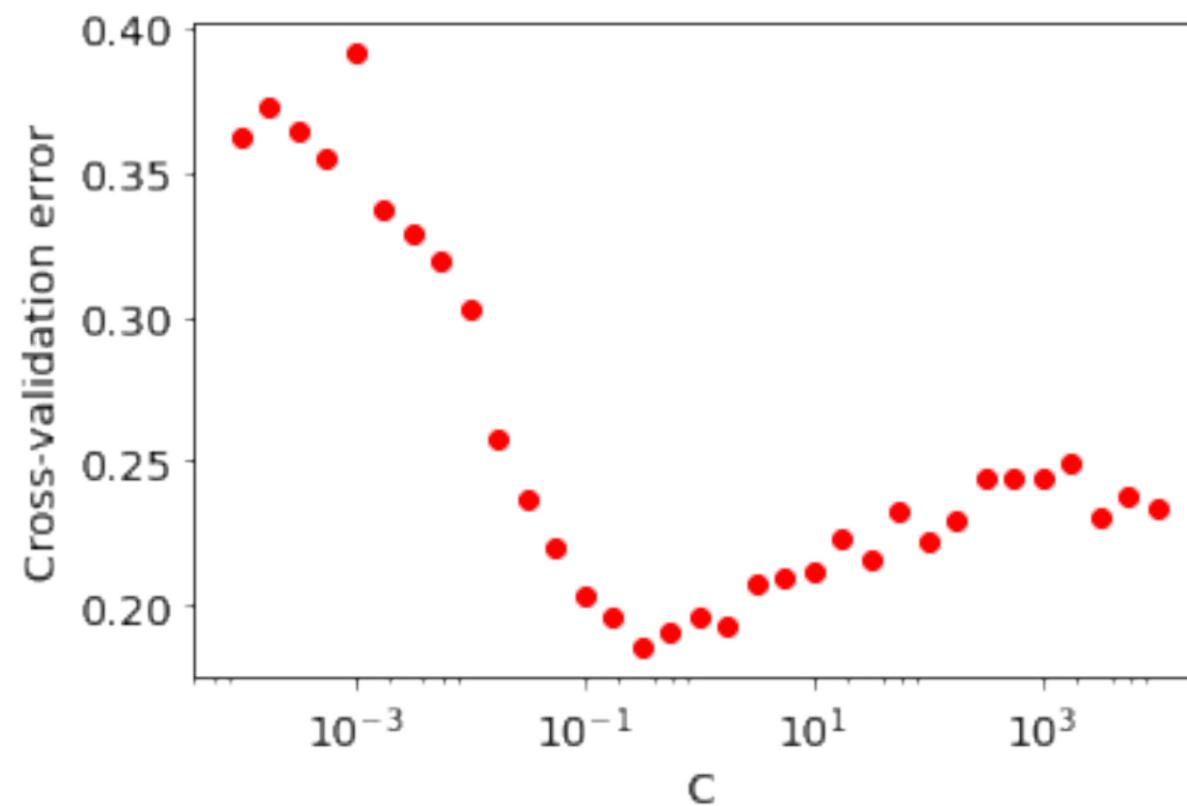
C : Trade-off entre la marge et les variables “slacks”.

$C = 0$: les erreurs ne sont pas pénalisées

$C = \infty$: aucune erreur n'est autorisée

Lorsque $C \rightarrow 0$, on obtient une marge plus grande.

Par validation croisée (cf TP) :



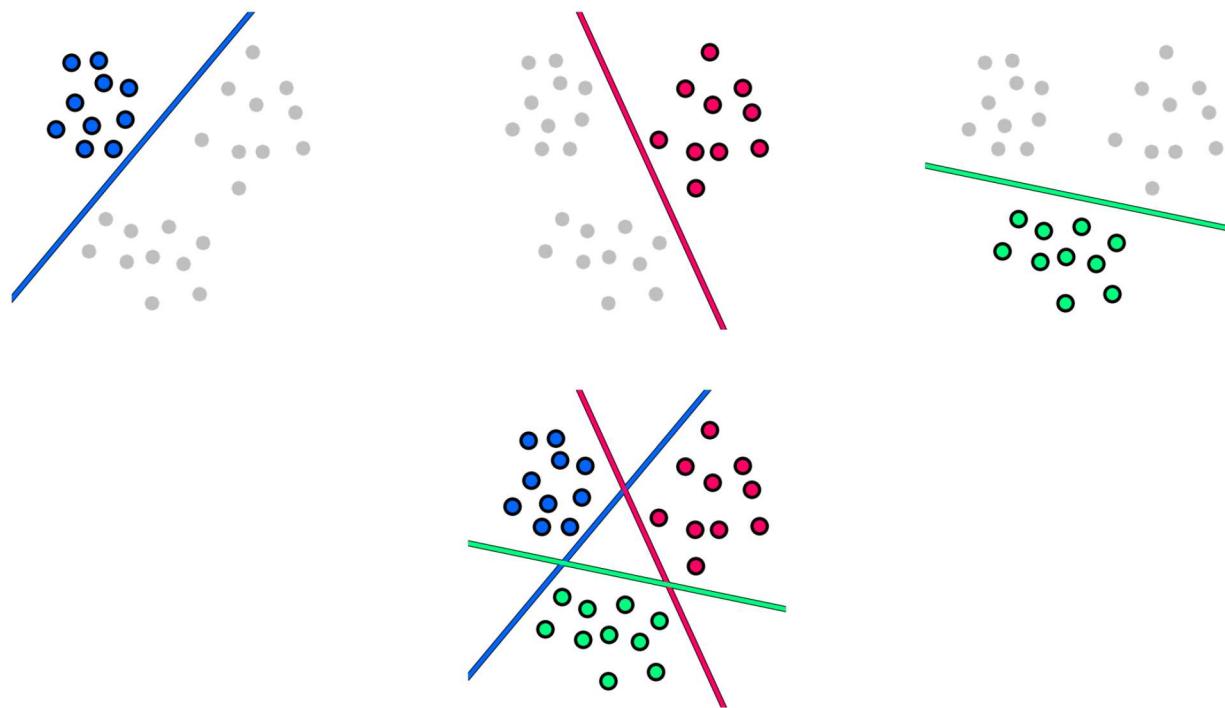
Classification linéaire Multiclasses



Contexte

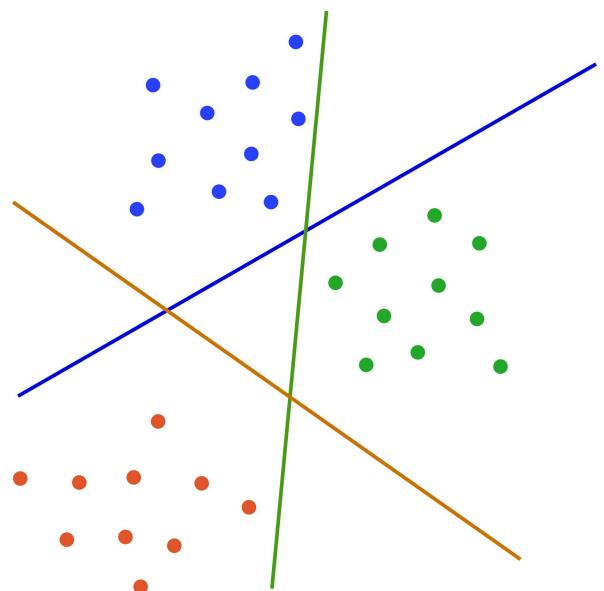
- **Espace des données** : $\mathcal{X} = \mathbb{R}^d$,
- **Ensemble des classes** : $\mathcal{Y} = \{1, \dots, k\}$
- **Modèle** : $w_1, \dots, w_k \in \mathbb{R}^d$ et $b_1, \dots, b_k \in \mathbb{R}$

Le classifieur w_j, b_j permet de distinguer les données de la classe j de toutes les autres.



Algorithme : 1 classe contre toutes

Puisque l'on sait classifier deux classes, alors pour déterminer par exemple la classe verte (+1), on assigne -1 à tous les autres points bleus et rouge et on utilise la classification binaire.

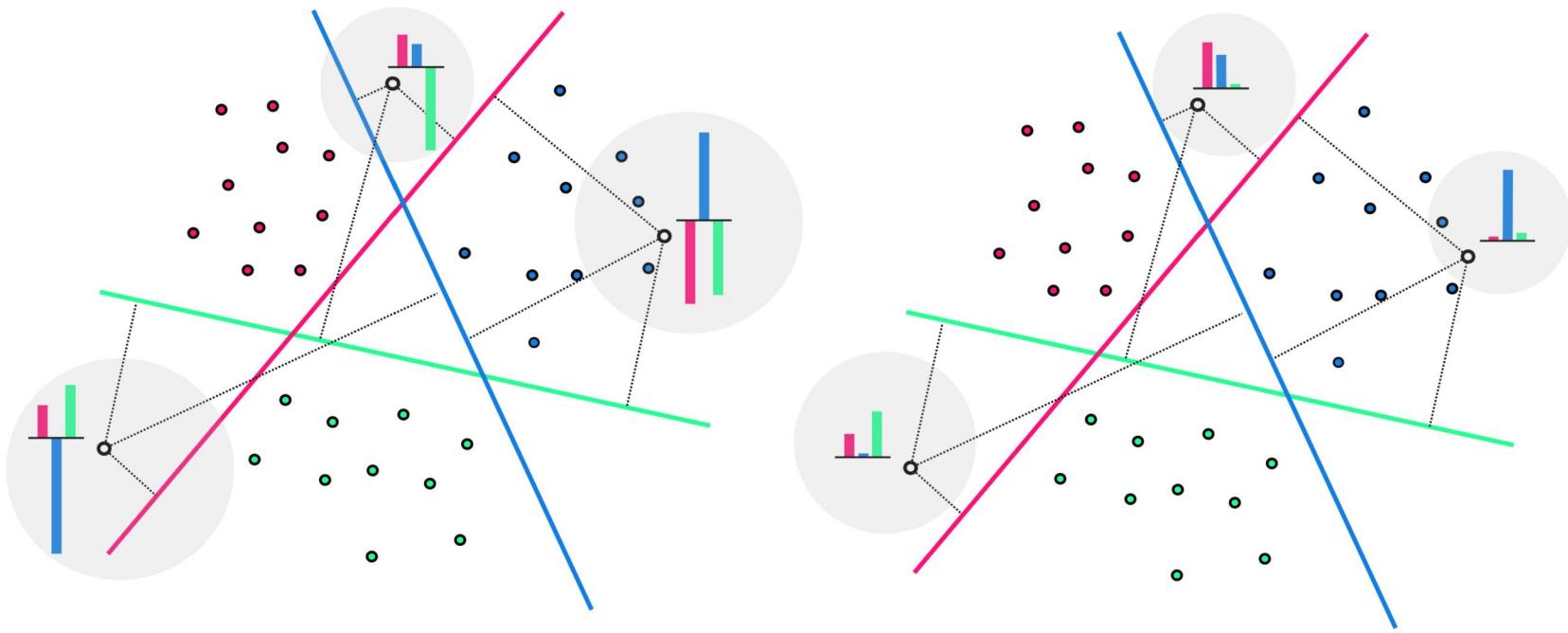


Cet algorithme ne peut pas mettre à jour simultanément tous les vecteurs w_1, \dots, w_k donc on verra plus loin un algorithme plus efficace.



La distance **signée** d'une donnée $x^{(i)}$ à la frontière du j ème classifieur est

$$\frac{w_j \cdot x^{(i)} + b_j}{\|w_j\|}, \quad (\text{normalisation : } \|w_j\| = 1).$$



La classe de la donnée $x^{(i)}$ est

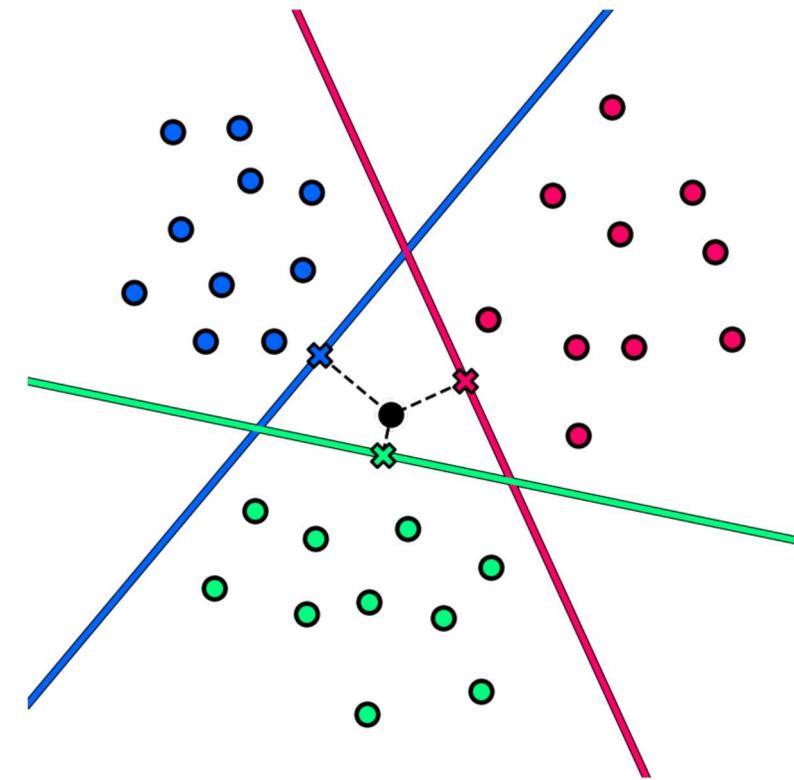
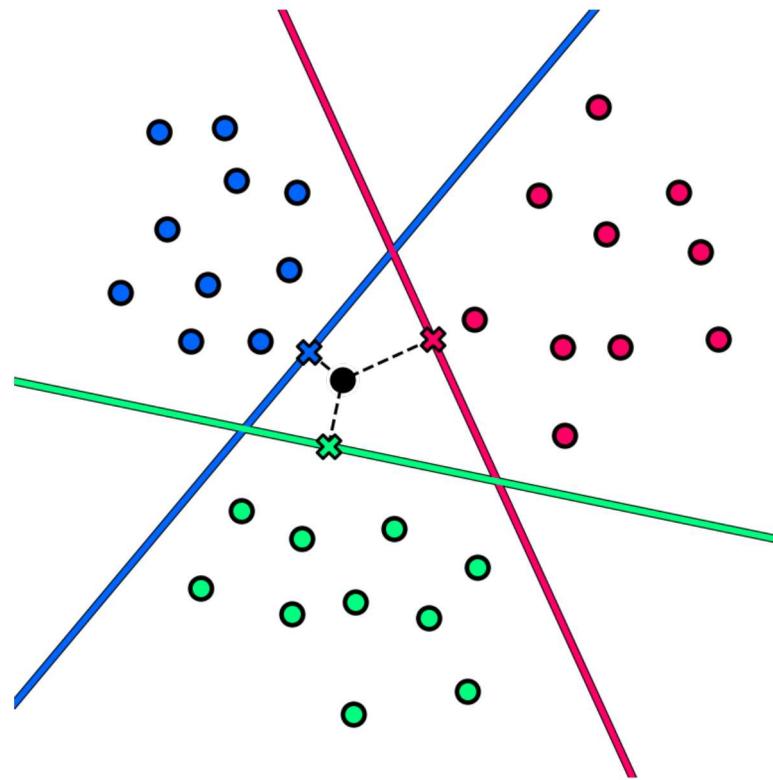
$$y^{(i)} = \operatorname{argmax}_{j=1, \dots, k} w_{y^{(j)}} \cdot x^{(j)} + b_{y^{(j)}}.$$



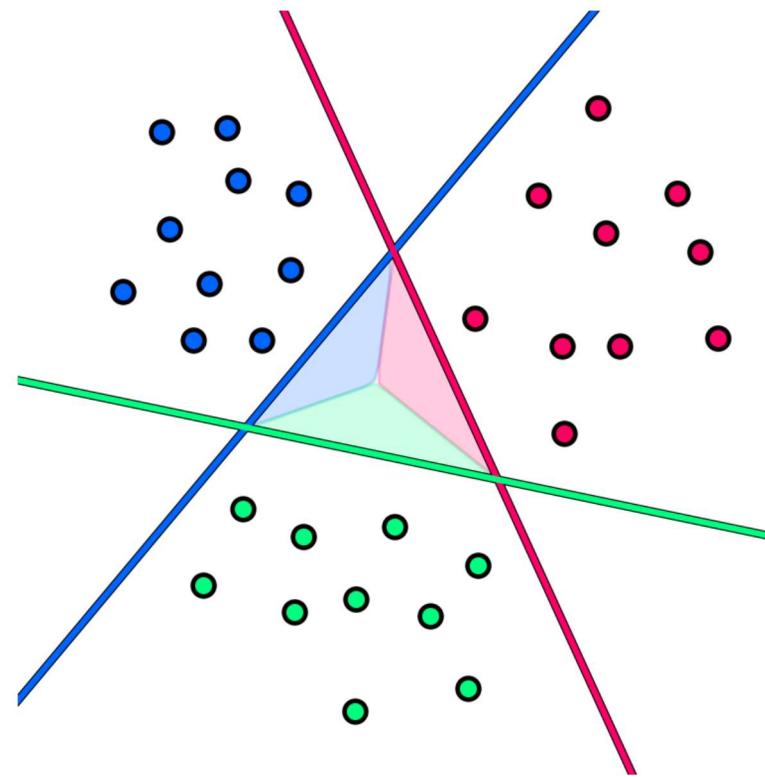
Remarque. Pour un point dans le demi-plan négatif d'un classifieur, la formule

$$y^{(i)} = \operatorname{argmax}_{j=1,\dots,k} w_{y^{(j)}} \cdot x^{(j)} + b_{y^{(j)}}$$

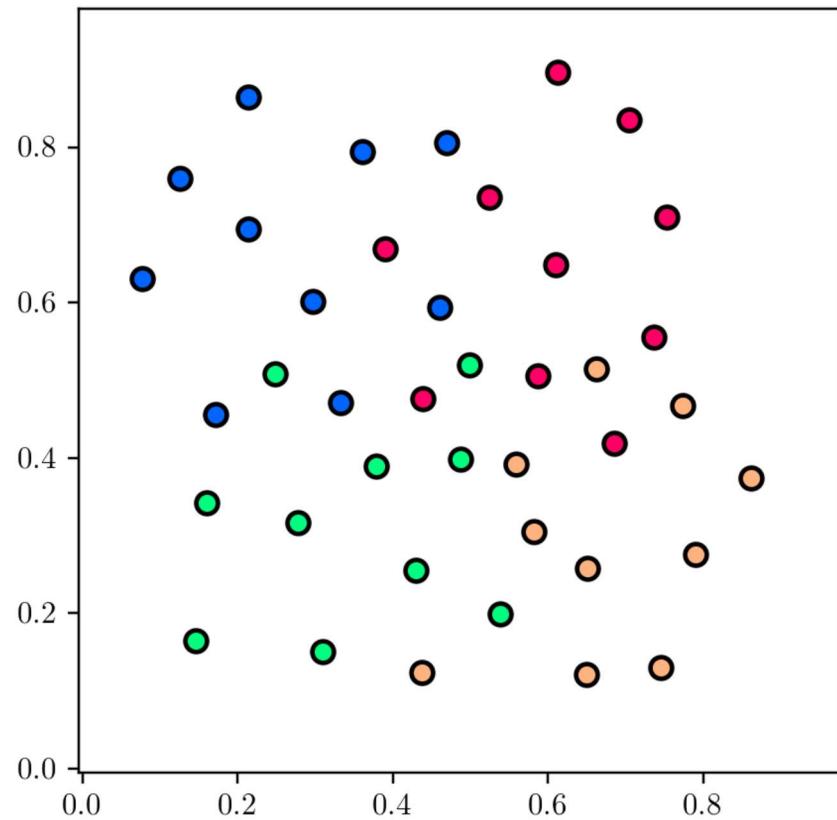
marche encore :



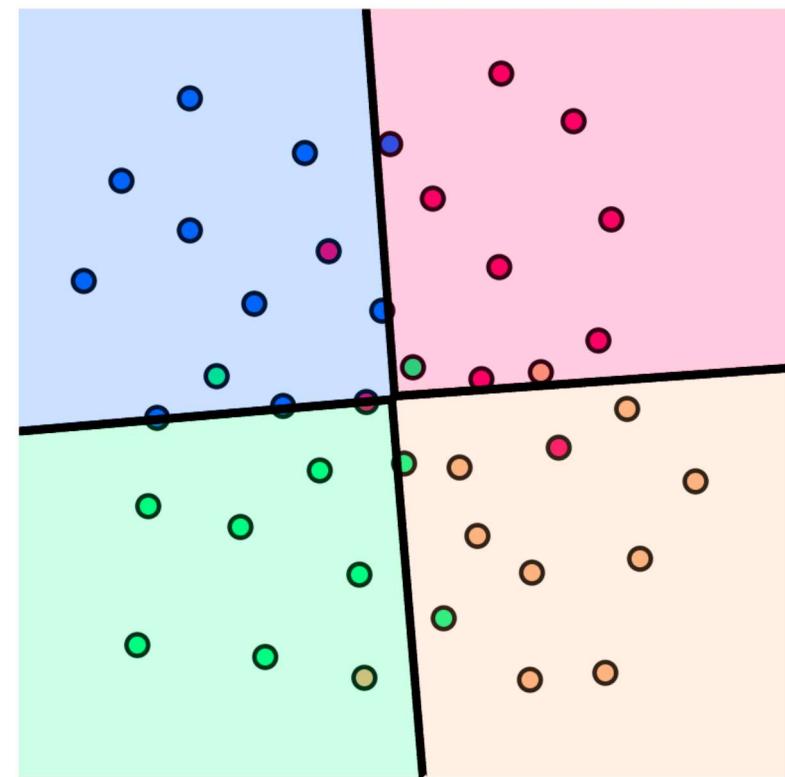
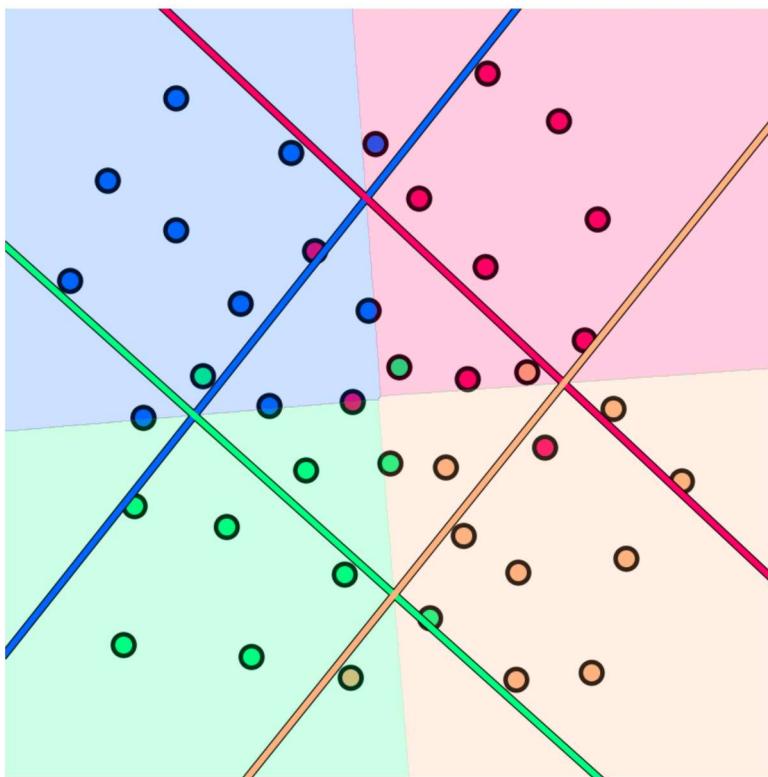
Une fois les hyperplans séparateurs déterminés, on en déduit les frontières de décision à partir des lieux de points équidistants à deux frontières.



Exemple



Exemple



Fonction Coût du Perceptron Multiclasse

Si $y^{(i)}$ est correcte alors

$$w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} = \max_{j=1,\dots,k} w_j \cdot x^{(i)} + b_j.$$



Fonction Coût du Perceptron Multiclasse

Si $y^{(i)}$ est correcte alors

$$w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} = \max_{j=1,\dots,k} w_j \cdot x^{(i)} + b_j.$$

Ainsi les n fonctions

$$g_i(w_1, \dots, w_k, b_1, \dots, b_k) = \left(\max_{j=1,\dots,k} w_j \cdot x^{(i)} + b_j \right) - \left(w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} \right) \geq 0$$

donnent une pénalisation pour les données $x^{(i)}$ incorrectes.



Fonction Coût du Perceptron Multiclasse

Si $y^{(i)}$ est correcte alors

$$w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} = \max_{j=1,\dots,k} w_j \cdot x^{(i)} + b_j.$$

Ainsi les n fonctions

$$g_i(w_1, \dots, w_k, b_1, \dots, b_k) = \left(\max_{j=1,\dots,k} w_j \cdot x^{(i)} + b_j \right) - \left(w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} \right) \geq 0$$

donnent une pénalisation pour les données $x^{(i)}$ incorrectes.

On forme alors la fonction de coût

$$g(w_1, \dots, w_k, b_1, \dots, b_k) = \frac{1}{n} \sum_{i=1}^n \max_{j=1,\dots,k} w_j \cdot x^{(i)} + b_j - \left(w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} \right)$$

à minimiser.



Algorithme du Perceptron pour la classification multiclasse

- **Prédiction** : Pour une donnée x , prédire la classe $\text{argmax}_j w_j \cdot x + b_j$.
- **Apprentissage** :



Algorithme du Perceptron pour la classification multiclasse

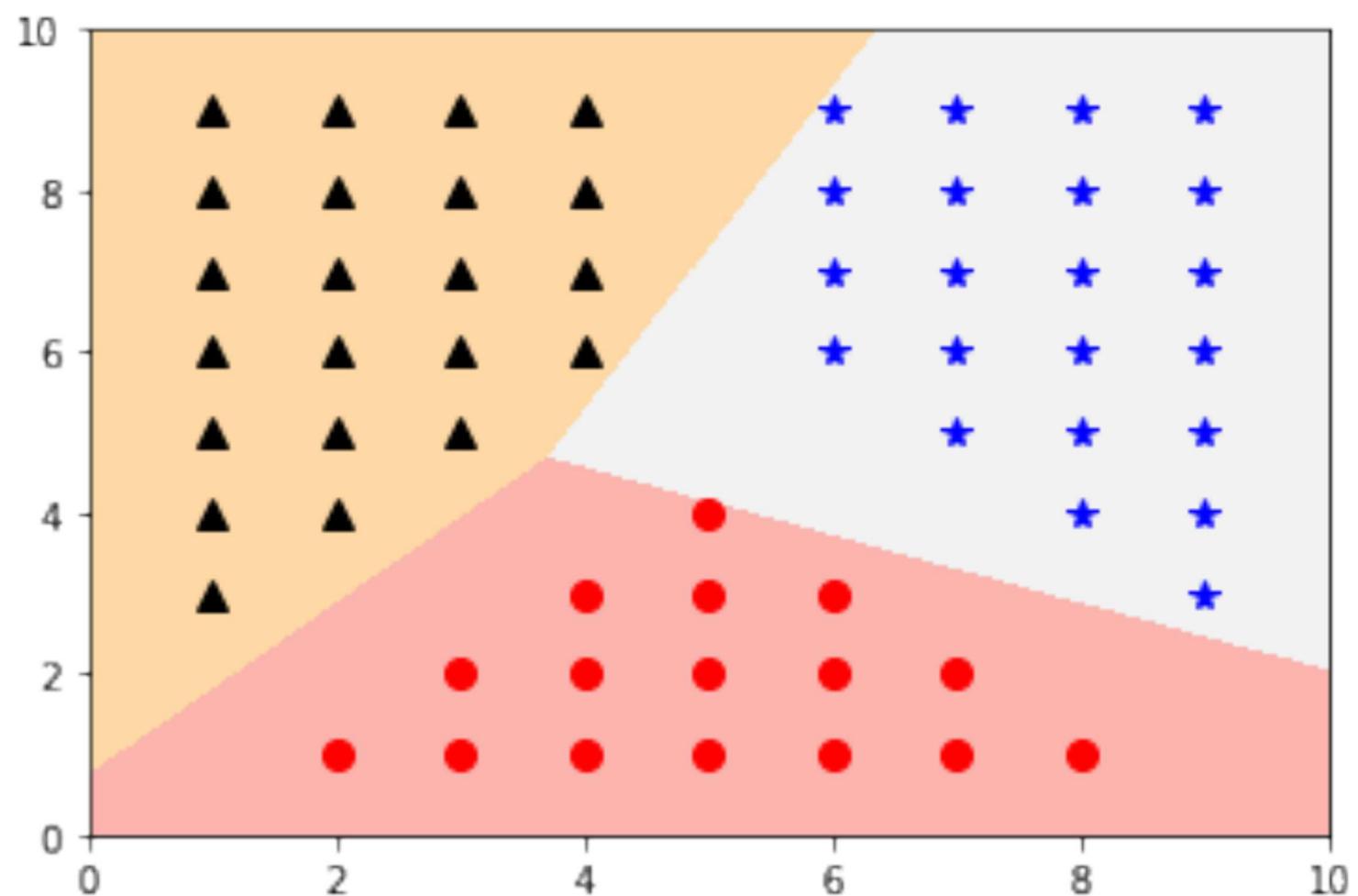
- **Prédiction** : Pour une donnée x , prédire la classe $\text{argmax}_j w_j \cdot x + b_j$.
- **Apprentissage** : On minimise la fonction coût du Perceptron étant donné des données d'entraînement $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$:
 - **Initialisation** $w_1 = \dots = w_k = 0$ et $b_1 = \dots = b_k = 0$
 - **Itérations** Tant qu'une donnée d'entraînement (x, y) est classée incorrectement
 - pour la classe correcte y , faire :

$$w_y \leftarrow w_y + x, \quad b_y \leftarrow b_y + 1$$

- pour la classe prédite \hat{y} , faire :

$$w_{\hat{y}} \leftarrow w_{\hat{y}} - x, \quad b_{\hat{y}} \leftarrow b_{\hat{y}} - 1$$





Problème d'optimisation du Perceptron Régularisé (**SVM**)

Comme pour le cas de la classification binaire, on régularise (unicité de la solution, problème bien posé ...) :

$$\begin{array}{ll} \min_{\substack{w_1, \dots, w_k, \\ b_1, \dots, b_k}} & \frac{1}{n} \sum_{i=1}^n \max_{j=1, \dots, k} w_j \cdot x^{(i)} + b_j - \left(w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} \right) \\ \text{sous} & \|w_j\|^2 = 1, \quad j = 1, \dots, k. \end{array} \quad (1)$$

et ce problème d'optimisation correspond à l'algorithme **SVM** de classification multiconcaves.



Autre formulation du problème SVM multilabels

- **Prédiction :** Pour une donnée x , prédire la classe $\text{argmax}_j w_j \cdot x + b_j$.
- **Apprentissage :** Étant donné des données d'entraînement $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$:



Autre formulation du problème SVM multilabels

- **Prédiction :** Pour une donnée x , prédire la classe $\operatorname{argmax}_j w_j \cdot x + b_j$.
- **Apprentissage :** Étant donné des données d'entraînement $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$:
On minimise pour $w_1, \dots, w_k \in \mathbb{R}^d, b_1, \dots, b_k \in \mathbb{R}, \xi \in \mathbb{R}^n$ le coût

$$\frac{1}{2} \sum_{i=1}^k \|w_i\|^2 + C \sum_{i=1}^n \xi_i$$

sous les contraintes

$$w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} - (w_y \cdot x^{(i)} + b_y) \geq 1 - \xi_i, \quad \forall i \text{ et } \forall y \neq y^{(i)}$$
$$\xi \succeq 0.$$



Autre formulation du problème SVM multilabels

- **Prédiction :** Pour une donnée x , prédire la classe $\operatorname{argmax}_j w_j \cdot x + b_j$.
- **Apprentissage :** Étant donné des données d'entraînement $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$:
On minimise pour $w_1, \dots, w_k \in \mathbb{R}^d, b_1, \dots, b_k \in \mathbb{R}, \xi \in \mathbb{R}^n$ le coût

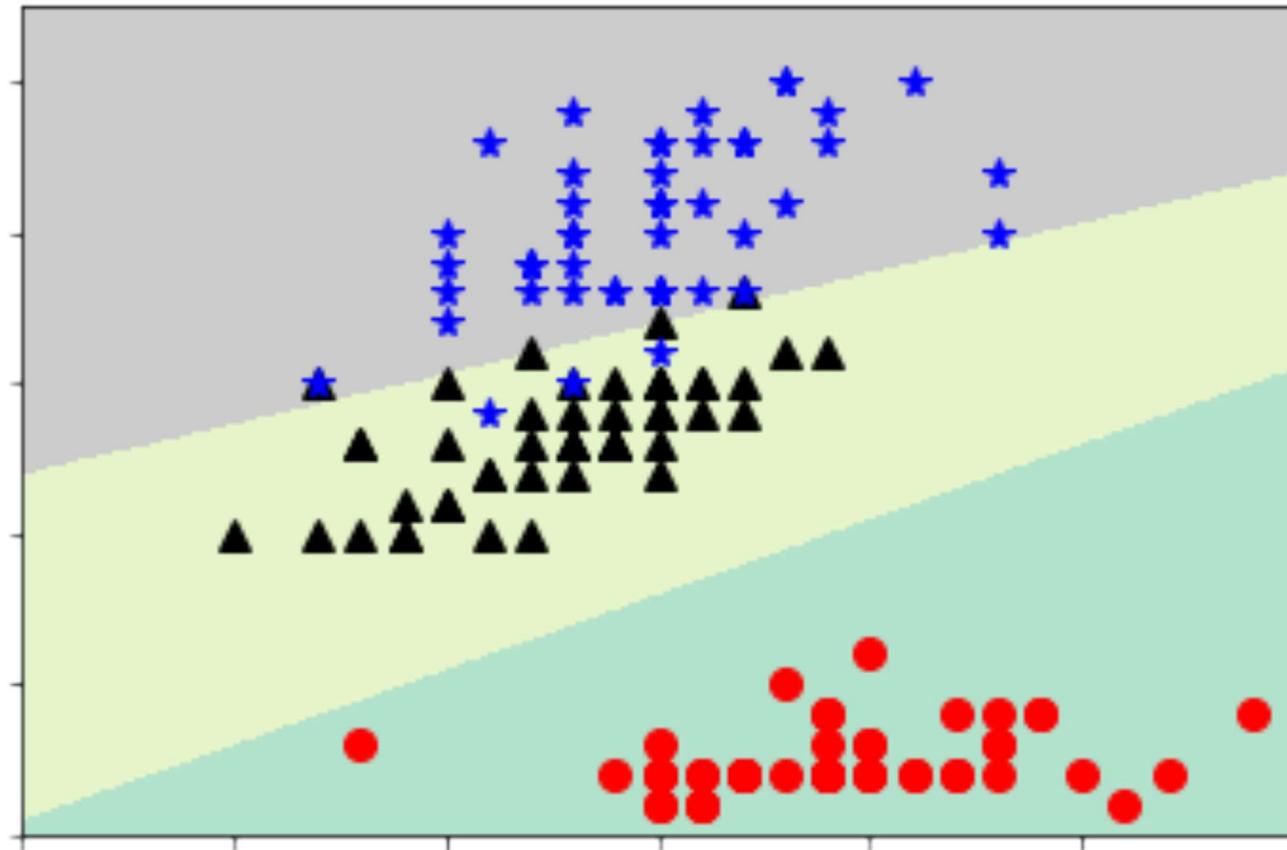
$$\frac{1}{2} \sum_{i=1}^k \|w_i\|^2 + C \sum_{i=1}^n \xi_i$$

sous les contraintes

$$w_{y^{(i)}} \cdot x^{(i)} + b_{y^{(i)}} - (w_y \cdot x^{(i)} + b_y) \geq 1 - \xi_i, \quad \forall i \text{ et } \forall y \neq y^{(i)}$$
$$\xi \succeq 0.$$

Exercice. Combien de contraintes possède ce problème d'optimisation convexe ?
Écrire son dual.





Exercice

Soient $P = \{A_1(-3, 2), A_2(-2, 2), A_3(-1, 2)\}$ et $N = \{A_4(1, 1), A_5(1, 0), A_6(2, 0)\}$ deux ensembles de \mathbb{R}^2 .

- ① Montrer graphiquement que les ensembles P et N sont strictement linéairement séparables par une droite et que la solution n'est pas unique.
- ② **Méthode Perceptron** : pour trouver un vecteur $w = (w_1, w_2, w_3) \in \mathbb{R}^3$ tel que

$$w \cdot \tilde{x} = w_1 x_1 + w_2 x_2 - w_3 > 0 \text{ si } x = (x_1, x_2) \in P$$
$$w \cdot \tilde{x} = w_1 x_1 + w_2 x_2 - w_3 < 0 \text{ si } x = (x_1, x_2) \in N,$$

avec la notation $\tilde{x} = (x_1, x_2, -1)$, nous utilisons la méthode itérative suivante : partant d'un vecteur $w(0)$ initial, le vecteur $w(t+1)$ se déduit de $w(t)$ à l'étape $t+1$ par les équations :

$$w(t+1) = w(t) + \tilde{x}, \text{ si } x \in P \text{ et } w \cdot \tilde{x} \leq 0$$
$$w(t+1) = w(t) - \tilde{x}, \text{ si } x \in N \text{ et } w \cdot \tilde{x} > 0$$
$$w(t+1) = w(t), \text{ sinon.}$$

