IA : Machine (Deep) Learning

Annexe : Perceptron

A. Zemmari

LaBRI - Université de Bordeaux

Rappel

Classification binaire

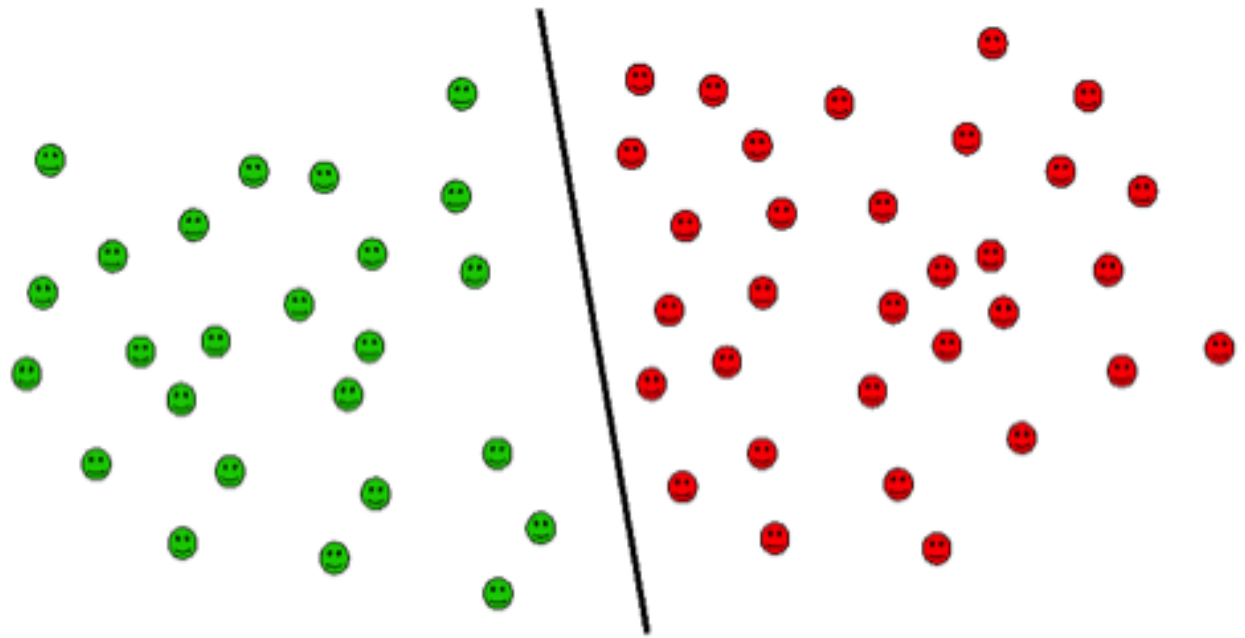
- Données : quelques éléments qui appartiennent à deux classes différentes
  - classe 1 (+1) et classe 2 (-1)
  - classe positive (+1) et classe négative (-1)
- Tâche : entrainer un classifieur sur ces données puis prédire la classe d'un nouvel élément
- Géométriquement : trouver une séparation entre les deux classes

Séparation Linéaire / Non Linéaire

Données séparables linéairement

Tous les points associés aux données peuvent être séparé s correctement par une frontière linéaire (hyperplan)

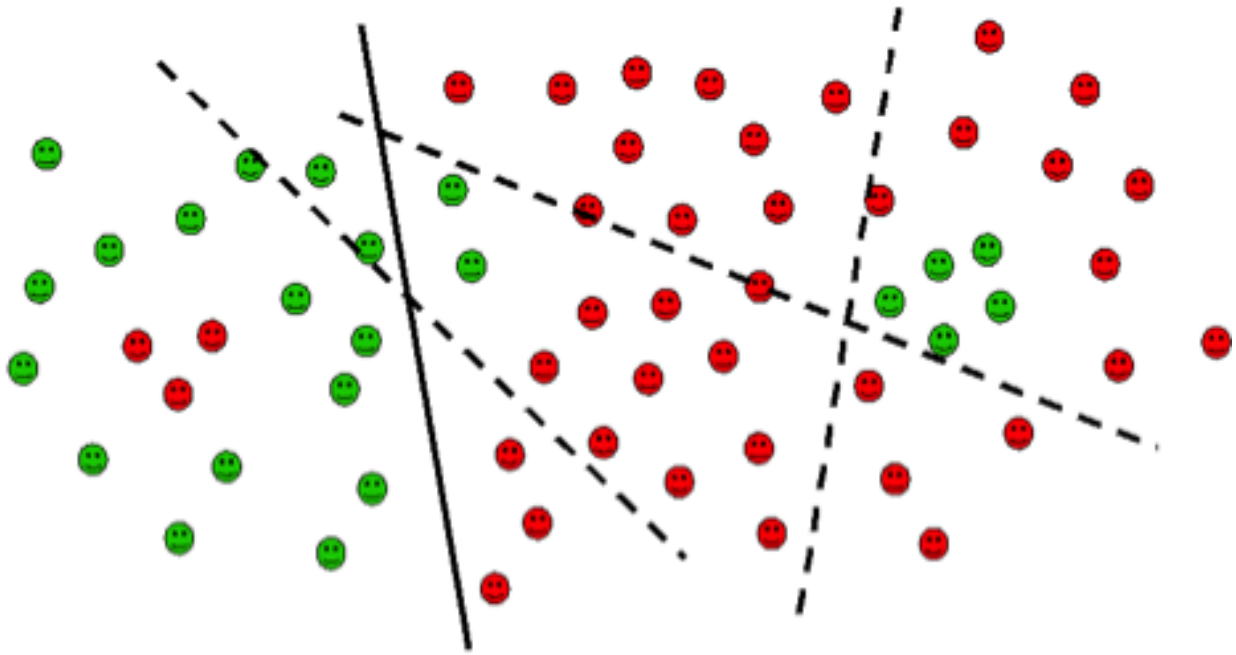
Exemple



Frontière de décision linéaire

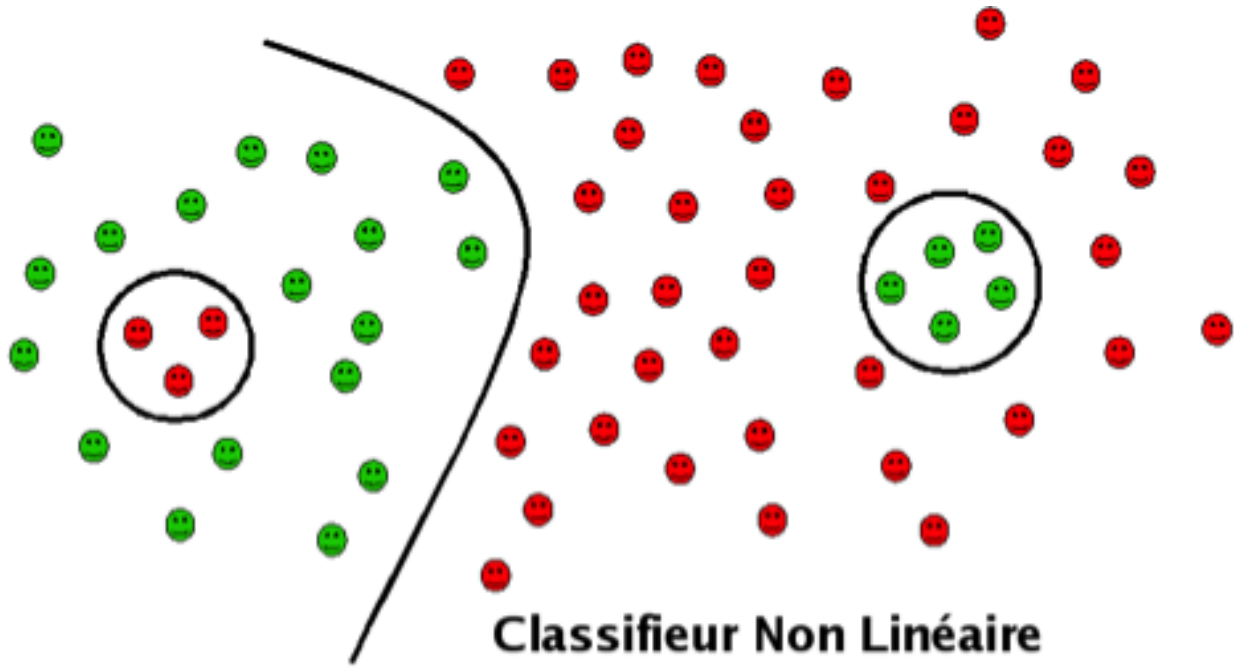
Données non séparables linéairement

Exemple : pas de séparation linéaire



Données non séparables linéairement

Exemple : pas de séparation linéaire



Algorithmes : Séparation Linéaire / Non Linéaire

Séparation Linéaire

- Données séparables Linéairement ou Non Linéairement ? réponse empirique
- Algorithmes Linéaires (algorithmes qui trouvent une frontière linéaire)
  - Quand on pense que les données sont linéairement séparables
  - Avantages : Simples, peu de paramètres à régler
  - Désavantages : les données dans un espace de grande dimension sont souvent non linéairement séparables
  - Exemples d'algorithmes : Perceptron, SVM

Algorithmes : Séparation Linéaire / Non Linéaire

Séparation Non Linéaire

- Non Linéaires
  - Quand les données sont non linéairement séparables
  - Avantages : Plus précis
  - Désavantages : Plus compliqués, plus de paramètres à régler
  - Exemple : méthodes à base de fonctions noyau
- Note : la distinction entre linéaire et non linéaire est valable pour la classification multi-classes

Algorithme du Perceptron

Perceptron : généralités

- Linéaire
- Classification Binaire
- En ligne (apprentissage séquentiel, une donnée à la fois)
- Apprentissage sur les erreurs
- Réseau de neurones à une couche

Algorithme du Perceptron

Perceptron : Principe

- On commence par initialiser aléatoirement la valeur des paramètres $w_1, w_2, \cdots, w_k$ et b.
- Ensuite, pour chaque exemple $(x_1, x_2, \cdots, x_k)$ de la base d'apprentissage, on compare la réponse fournie à la réponse attendue. Si elles sont égales, on ne modifie pas les poids. En revanche si la réponse fournie est 0 alors que celle attendue est 1, on ajoute 1 aux poids des neurones d'entrée activés : ceci a pour effet d'accroître leur influence. A contrario, si la réponse fournie est 1 alors que celle attendue est 0, on retire 1 aux poids des neurones d'entrée activés.
- L'algorithme d'apprentissage se poursuit jusqu'à ce que tous les éléments de la base d'apprentissage soient étudiés sans qu'aucun poids ne soient modifiés.

Algorithme du Perceptron

Algorithme linéaire simple

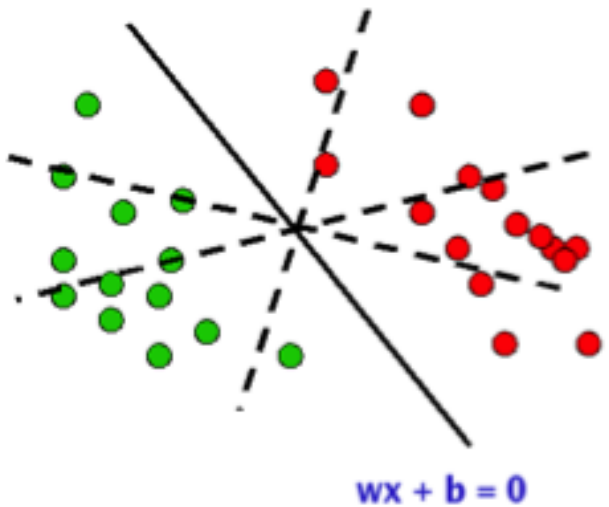
- Données : $\{(x_i, y_i)_{i=1,\cdots,n}\}$
- $x_i \in \mathbb{R}^d$ (x_i est un vecteur dans un espace de de dimension d)
  $\rightarrow$ vecteur de descripteurs
- $y_i \in \{-1, +1\} \Leftarrow$ étiquette de la classe
- Question
  - Trouver une frontière linéaire : $wx + b$ (équation de l'hyperplan) telle que la règle de classification associée donne une probabilité d'erreur minimale
  - règle de classification :
    $y = signe(wx + b)$, ce qui signifie :

$$\begin{cases} \text{si} & wx + b > 0 & \text{alors} & y = +1 \\ \text{si} & wx + b < 0 & \text{alors} & y = -1 \end{cases}$$

Algorithme du Perceptron

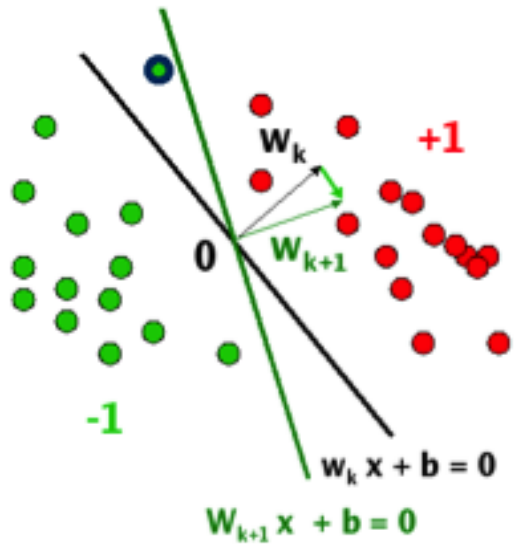
Principe

- Trouver un hyperplan $(w, b) \in \mathbb{R}^{d+1}$ qui classe aussi bien que possible les données (points)
- Progressivement : un point à la fois, en modifiant les poids si nécessaire



Algorithme du Perceptron

- Initialisation : $w_1 = 0$
- Mise à jour des poids
  pour chaque point x
  si $(classe(x) \neq decision(x, w))$ **alors**
  $w_{k+1} \leftarrow w_k + y_i x_i$
  $k \leftarrow k + 1$
  sinon
  $w_{k+1} \leftarrow w_k$
  fin si
- Fonction $decision(x, w)$
  si $(wx + b > 0)$ **alors**
  retourner $+1$
  sinon
  retourner -1
  fin si



Algorithme du Perceptron

Analyse

- Progressif : s'adapte toujours aux nouvelles données
- Avantages
  - Simple et efficace
  - Garantie d'apprendre un problème linéairement séparable (convergence, optimum global)
- Limitations
  - Seulement séparations linéaires
  - Converge seulement pour données séparables
  - Pas très efficace dès qu'il y a trop de descripteurs

Séparateur à vastes marges (SVM)

- Une autre famille d'algorithmes linéaires
- intuition (Vapnik, 1965)
- Si les classes sont linéairement séparables :
  - Séparer les données
  - Mettre un hyperplan "loin" des données : marge large
  - résultats statistiques garantis
  - bonne généralisation

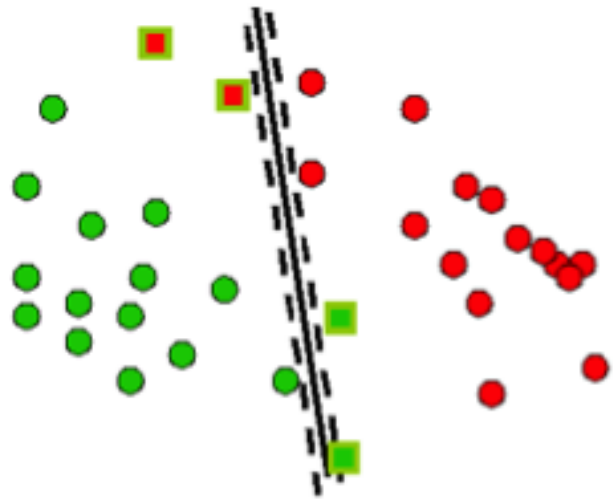


Figure: Mauvais

Séparateur à vastes marges (SVM)

- Une autre famille d'algorithmes linéaires
- intuition (Vapnik, 1965)
- Si les classes sont linéairement séparables :
  - Séparer les données
  - Mettre un hyperplan "loin" des données : marge large
  - résultats statistiques garantis
  - bonne généralisation

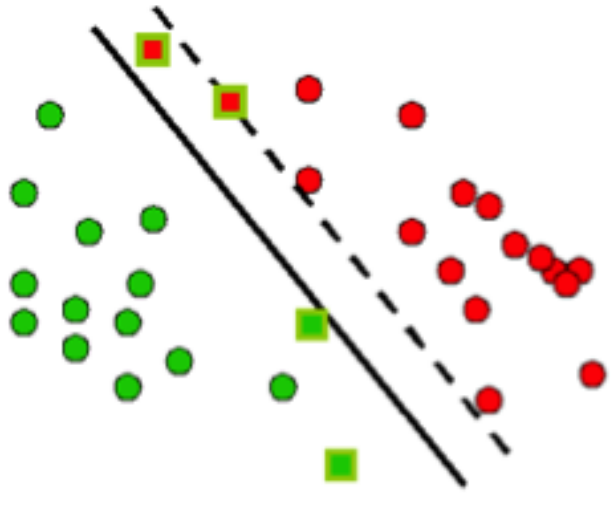
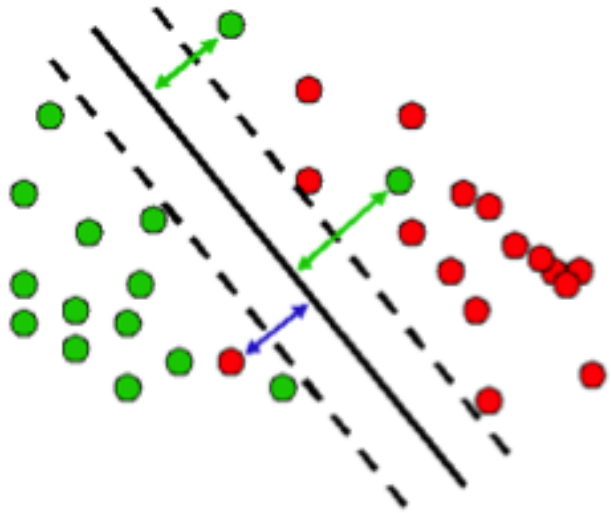


Figure: Bon $\Rightarrow$ Classifieur à marge maximale.

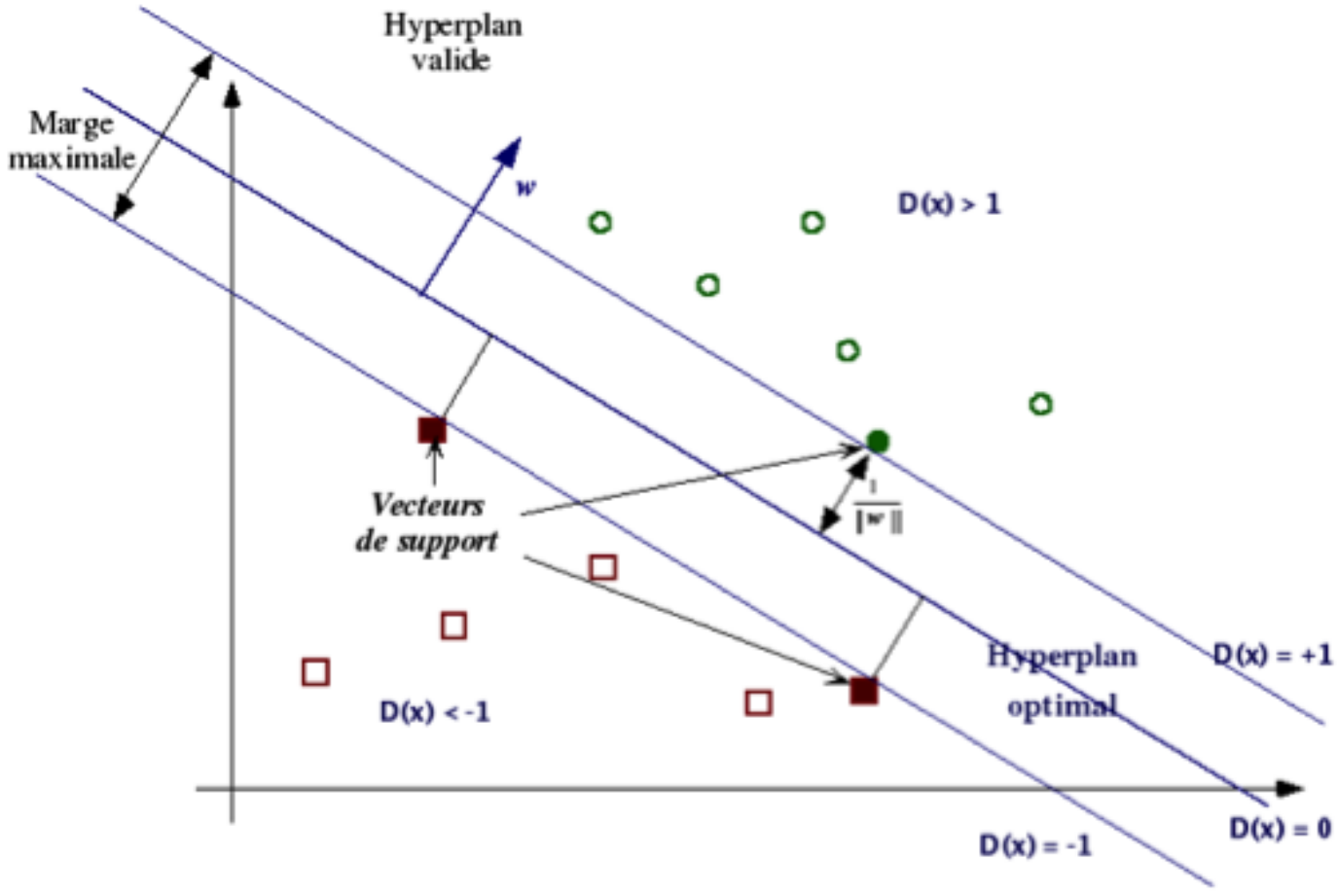
Séparateur à vastes marges (SVM)

- Si non séparable linéairement
  - Permettre quelques erreurs
  - Essayer encore de placer un hyperplan "loin" de chaque classe



Hyperplan de plus vaste marge

Optimiser la marge



Hyperplan de plus vaste marge

Optimiser la marge

- La distance d'un point à l'hyperplan est : $\frac{|wx+b|}{||w||}$
- L'hyperplan optimal est celui pour lequel la distance aux points les plus proches est maximale. La marge entre les deux classes vaut $\frac{2}{||w||}$.
- Maximiser la marge revient donc à minimiser $||w||$ sous ?les contraintes :

$$\begin{cases} min\frac{||w||^2}{2} \\ \forall i,\ y_i(wx_i + b) \geq 1 \end{cases}$$

SVM : un problème d'optimisation quadratique

Expression primaire

- Il faut donc déterminer w et b minimisant :

$$\frac{||w||^2}{2}$$

  afin de maximiser le pouvoir de généralisation.

- sous les contraintes (hyperplan séparateur):

$$y_i(wx_i + b) \geq 1, \ \forall i \in [1..n]$$

La résolution de ce problème n'est pas au programme de ce cours. L'idée est plutôt de donner l'intuition et expliquer comment la méthode marche ...