

Intelligence Artificielle pour l'Analyse de Données

SVM & Perceptron

Akka Zemmari

`zemmari@u-bordeaux.fr`

LaBRI, Université de Bordeaux

2019-2020

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 : entraîner un classifieur sur ces données puis prédire la classe d'un nouvel élément
- ▶ Géométriquement : trouver une séparation entre les deux classes

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** : entraîner un classifieur sur ces données puis prédire la classe d'un nouvel élément
- ▶ **Géométriquement** : trouver une séparation entre les deux classes

Rappel

Classification binaire

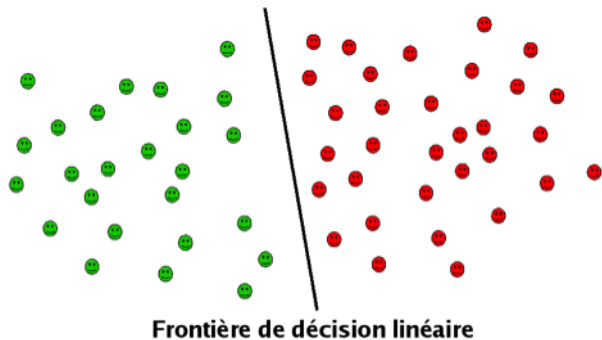
- ▶ Données : quelques éléments qui appartiennent à deux classes différentes
 - ▶ classe 1 (+1) et classe 2 (-1)
 - ▶ classe positive (+1) et classe négative (-1)
- ▶
- ▶ Tâche : entraîner un classifieur sur ces données puis prédire la classe d'un nouvel élément
- ▶ Géométriquement : trouver une séparation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

Séparation Linéaire / Non Linéaire

Données linéairement séparables

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

Exemple

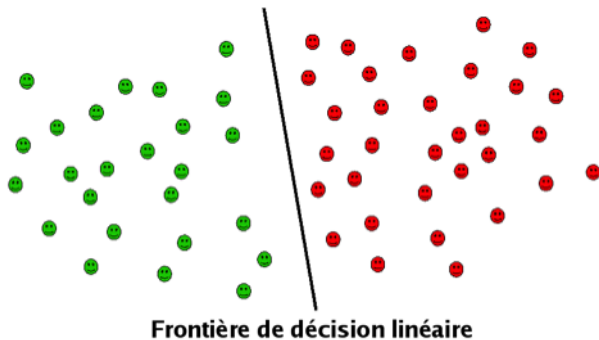


Séparation Linéaire / Non Linéaire

Données linéairement séparables

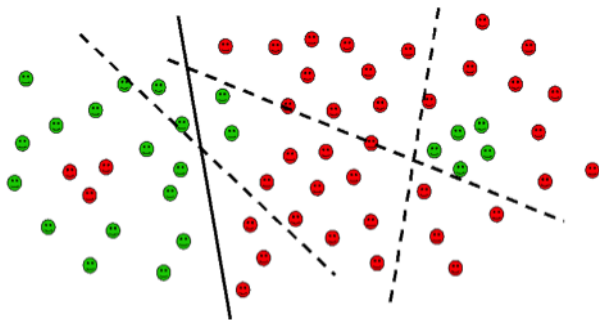
Tous les points associés aux données peuvent être séparés correctement par une frontière linéaire (hyperplan)

Exemple



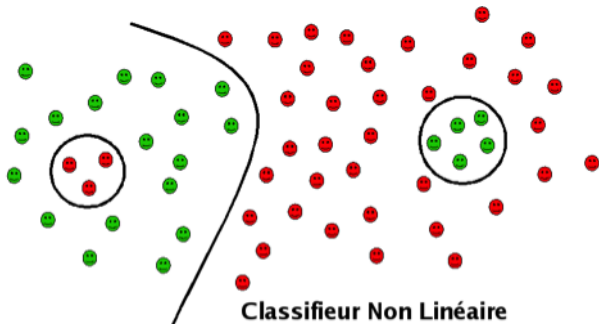
Séparation Linéaire / Non Linéaire

Exemple : pas de séparation linéaire



Séparation Linéaire / Non Linéaire

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 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 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 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 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 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

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

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

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

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

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

Algorithme linéaire simple

- ▶ Données : $\{(x_i, y_i)_{i=1, \dots, n}\}$
- ▶ $x_i \in \mathbb{R}^d$ (x_i est un vecteur dans un espace de dimension d)
⇒ vecteur de features
- ▶ $y_i \in \{-1, +1\}$ ⇒ 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 :
 $y = \text{signe}(wx + b)$, 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

Algorithme linéaire simple

- ▶ **Données** : $\{(x_i, y_i)_{i=1, \dots, n}\}$
- ▶ $x_i \in \mathbb{R}^d$ (x_i est un vecteur dans un espace de dimension d)
⇒ vecteur de features
- ▶ $y_i \in \{-1, +1\}$ ⇒ 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 :
 $y = \text{signe}(wx + b)$, 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

Algorithme linéaire simple

- ▶ Données : $\{(x_i, y_i)_{i=1, \dots, n}\}$
- ▶ $x_i \in \mathbb{R}^d$ (x_i est un vecteur dans un espace de dimension d)
⇒ vecteur de features
- ▶ $y_i \in \{-1, +1\}$ ⇒ 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 :
 $y = \text{signe}(wx + b)$, 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

Algorithme linéaire simple

- ▶ Données : $\{(x_i, y_i)_{i=1, \dots, n}\}$
- ▶ $x_i \in \mathbb{R}^d$ (x_i est un vecteur dans un espace de dimension d)
 \Rightarrow vecteur de features
- ▶ $y_i \in \{-1, +1\} \Rightarrow$ 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 :
 $y = \text{signe}(wx + b)$, 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

Algorithme linéaire simple

- ▶ Données : $\{(x_i, y_i)_{i=1, \dots, n}\}$
- ▶ $x_i \in \mathbb{R}^d$ (x_i est un vecteur dans un espace de dimension d)
⇒ vecteur de features
- ▶ $y_i \in \{-1, +1\}$ ⇒ 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 :
 $y = \text{signe}(wx + b)$, 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

Algorithme linéaire simple

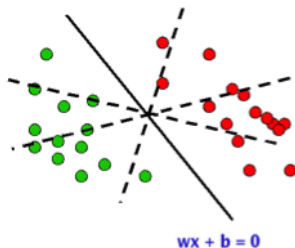
- ▶ Données : $\{(x_i, y_i)_{i=1, \dots, n}\}$
- ▶ $x_i \in \mathbb{R}^d$ (x_i est un vecteur dans un espace de dimension d)
 \Rightarrow vecteur de features
- ▶ $y_i \in \{-1, +1\} \Rightarrow$ étiquette de la classe
- ▶ Question
 - ▶ Trouver une frontière linéaire : $wx + b$ (équation de l'hyperplan)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

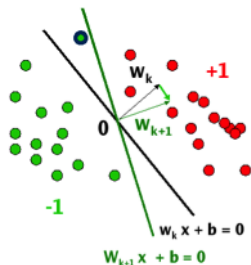
Principe

- ▶ 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



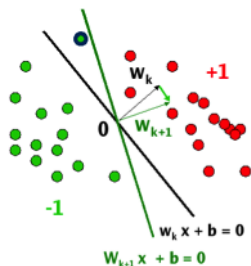
Algorithme du Perceptron

- ▶ 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**



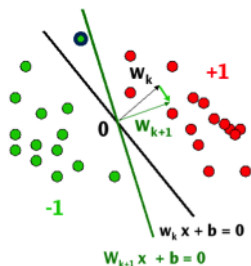
Algorithme du Perceptron

- ▶ 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**



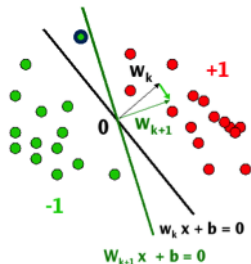
Algorithme du Perceptron

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



Algorithme du Perceptron

- ▶ Initialisation : $w_1 = 0$
- ▶ Mise à jour des poids pour chaque point x
si ($\text{classe}(x) \neq \text{decision}(x, w)$) **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 $\text{decision}(x, w)$
si ($w x + b > 0$) **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 features

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 features

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 features

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 features

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
 - ▶ bonne généralisation

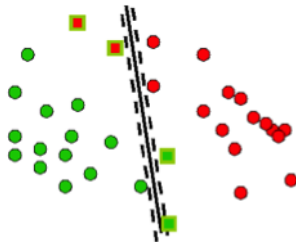


Figure: Mauvais

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
 - ▶ bonne généralisation

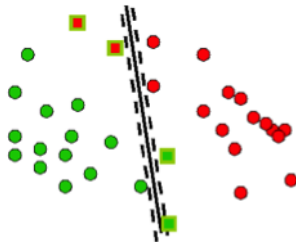


Figure: Mauvais

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
 - ▶ bonne généralisation

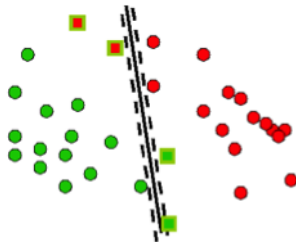


Figure: Mauvais

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
 - ▶ bonne généralisation

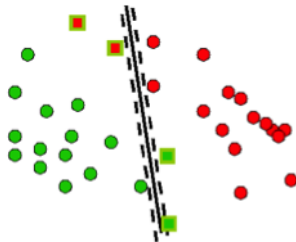


Figure: Mauvais

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
 - ▶ bonne généralisation

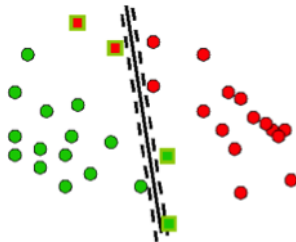


Figure: Mauvais

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
 - ▶ bonne généralisation

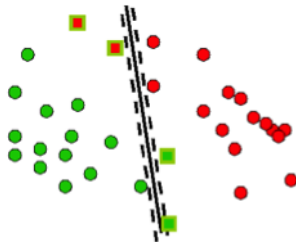


Figure: Mauvais

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
 - ▶ bonne généralisation

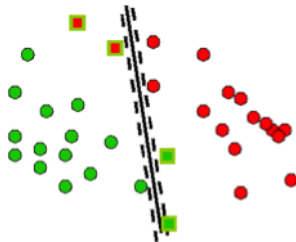


Figure: Mauvais

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
 - ▶ bonne généralisation

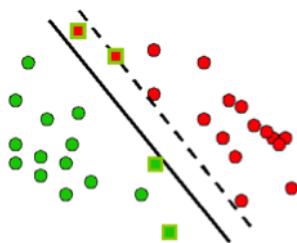
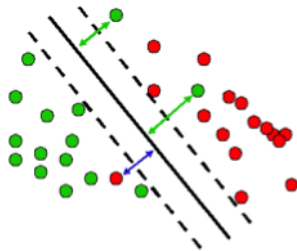


Figure: Bon \Rightarrow Classifieur à marge maximale.

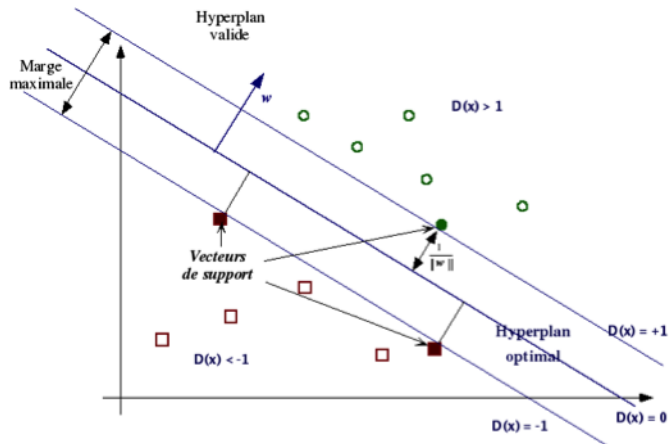
Séparateur à vastes marges (SVM)

- ▶ 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 ...

Données non linéairement séparables : SVM à noyau (kernel trick)

Intuition : un exemple.

Soit les données suivantes :



Données non linéairement séparables : SVM à noyau (kernel trick)

Intuition : un exemple.

Appliquons aux données la transformation suivante :

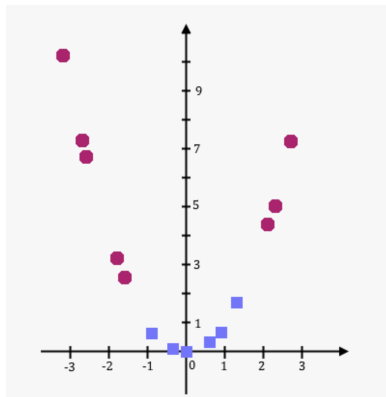
$$\phi(x) = \left(\frac{x - 150}{10}, \left(\frac{x - 150}{10} \right)^2 \right)$$

On passe donc de la dimension 1 à une dimension 2.

Données non linéairement séparables : SVM à noyau (kernel trick)

Intuition : un exemple.

Soit les données suivantes :



Données non linéairement séparables : SVM à noyau (kernel trick)

Noyaux couramment utilisés :

- ▶ Noyau polynomial.
- ▶ Noyau gaussien.
- ▶ Noyau laplacien.
- ▶ Noyau rationnel.