

Cours Optimisation et Apprentissage

EMIASD

Mathieu Lacroix

Outline

- 1 Problème d'optimisation
- 2 Problème d'optimisation linéaire
- 3 Problème d'optimisation linéaire en variables mixtes
- 4 Support Vector Machine

Outline

- 1 Problème d'optimisation
- 2 Problème d'optimisation linéaire
- 3 Problème d'optimisation linéaire en variables mixtes
- 4 Support Vector Machine

Problème d'optimisation

Forme générale

$$\min f(x) \quad (1)$$

$$h(x) = 0 \quad (2)$$

$$g(x) \leq 0 \quad (3)$$

$$x \in S \quad (4)$$

- f , g et h peuvent être de forme linéaire, convexe ou générale
- $S = \mathbb{R}^n$ ou $S = \mathbb{R}^n \times \mathbb{Z}^k$

Exemples

Problème d'optimisation quadratique continu

$$\min x^2 + 3y^2 \quad (5)$$

$$x + y = 1 \quad (6)$$

$$2xy \leq 4x + 2 \quad (7)$$

$$x, y \in \mathbb{R} \quad (8)$$

Problème d'optimisation linéaire mixte

$$\min 3x + 4y \quad (9)$$

$$x + y \geq 1 \quad (10)$$

$$2x + 3y \leq 6 \quad (11)$$

$$x, y \geq 0 \quad (12)$$

$$y \text{ entier} \quad (13)$$

Classification et solveurs

- Problèmes d'optimisation linéaire (mixte) : SCIP, CBC, HiGHS, Gurobi*
- Problème d'optimisation non linéaire : SCIP, IPOPT (continu), XPRESS*, KNITRO*

* solveur commercial

Intérêt

Plutôt que de développer des méthodes de résolution, on formule les problèmes et on utilise un solveur.

Outline

- 1 Problème d'optimisation
- 2 Problème d'optimisation linéaire
- 3 Problème d'optimisation linéaire en variables mixtes
- 4 Support Vector Machine

Exemple: production de yaourt

Une entreprise de fabrication de yaourts souhaite produire des yaourts à la fraise. Production possible de deux types de yaourts : allégé (40 euros/L) et normal (50 euros/L).

	Fraise	Lait	Sucre
Allégé	2kg	1L	
Normal	1kg	2L	1kg

Étant donné que l'entreprise possède 800 kilos de fraises, 700 litres de lait et 300 kilos de sucre, combien de litres de yaourt allégé et normal doit-elle fabriquer pour maximiser son revenu ?

Résolution d'un problème d'optimisation linéaire

Deux variables \Rightarrow résolution graphique !

Cas général: algorithme du simplexe

- Partir d'une solution de base réalisable (sommet du polyèdre)
- Parcourir les sommets jusqu'à arriver à une solution optimale
 - On passe d'un sommet à un de ses voisins (pivot)
 - On choisit un voisin avec un coût supérieur ou égal au sommet courant.

Complexité

- Algorithme du simplexe :
 - exponentiel dans le pire cas
 - très rapide en pratique
- Il existe des algorithmes polynomiaux d'optimisation linéaire (ex: méthode des points intérieurs).
- Question ouverte : existe-t-il un algorithme fortement polynomial ?*

** Une des 18 questions/challenges du 21ème siècle selon le médaille Fields Stephen Smale !*

Outline

- 1 Problème d'optimisation
- 2 Problème d'optimisation linéaire
- 3 Problème d'optimisation linéaire en variables mixtes
- 4 Support Vector Machine

Introduction

Forme générale

- Objective linéaire à optimiser
- Contraintes linéaires
- Certaines variables ne peuvent prendre que des valeurs entières

$$\max c^T x \quad (14)$$

$$Ax \leq b \quad (15)$$

$$x \in \mathbb{R}^n \times \mathbb{Z}^{n'} \quad (16)$$

Complexité

L'optimisation linéaire en variables mixte est NP-difficile.

Exemple

$$\max 3x_1 + 8x_2 \quad (17)$$

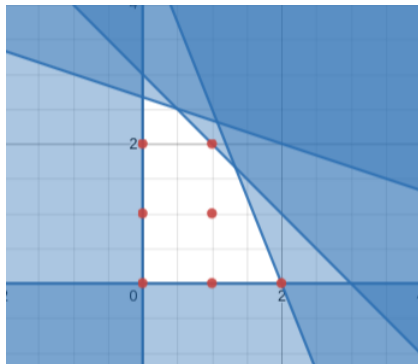
$$x_1 + x_2 \leq 3 \quad (18)$$

$$5x_1 + 2x_2 \leq 10 \quad (19)$$

$$x_1 + 3x_2 \leq 8 \quad (20)$$

$$x_1, x_2 \geq 0 \quad (21)$$

$$x_1, x_2 \text{ entiers} \quad (22)$$



Exemple

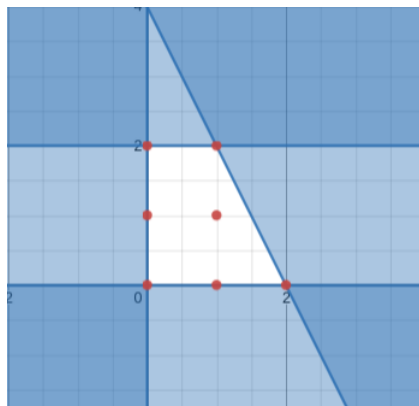
$$\max 3x_1 + 8x_2$$

$$2x_1 + x_2 \leq 4$$

$$x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \text{ entiers}$$



Énumération explicite

Quand le domaine est borné, le nombre de solutions est borné. On peut donc énumérer toutes les solutions et prendre la meilleure.

- Variables binaires uniquement
- Test de 10^9 solutions par seconde

n	Temps d'exécution de l'algorithme
10	1 microseconde
20	1 milliseconde
30	1 seconde
40	18.3 minutes
50	13 jours
60	36,5 années
70	374,3 siècles

⇒ Impossible !

Algorithme de séparation et évaluation (cas binaire)

- Méthode générique
- Énumération implicite des solutions
- Utilise un arbre de branchements (divise le problème en sous-problèmes)

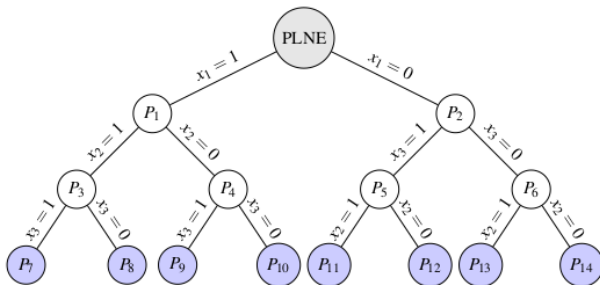


FIGURE 2.2 – Arbre de branchements pour 3 variables

Algorithme de séparation et évaluation (cas binaire)

Important

Le but de l'algorithme de séparation et évaluation est de ne pas explorer tout l'arbre !

- Quelle est la meilleure solution obtenue pour l'instant ?
- Peut-on espérer obtenir mieux (une meilleure solution) en explorant cette partie de l'arbre ?

Algorithme de séparation et évaluation (cas binaire)

- ① $L = \{S_0\}$, $\text{bestSol} = -\infty$
- ② while $L \neq \emptyset$
 - ① $S = \text{retirer sommet}(L)$
 - ② $x^{PL} = \text{relaxation linéaire}(S)$
 - ③ si x^{PL} entier
 - ① Mettre à jour bestSol (et sauvegarder x^{pl}) si $c^T x^{PL}$ est supérieur à bestSol .
 - ④ sinon si $c^T x^{PL} > \text{bestSol}$
 - ① Choisir une variable fractionnaire x_i
 - ② Créer S' et S'' correspondant à S avec $x_i = 0$ et $x_i = 1$
 - ③ Ajouter S' et S'' dans L

Spécifications

- Choix du sommet à explorer
- Choix de la variable sur laquelle brancher

Variables entières

On branche sur $x_i \geq \lceil x_i^{PL} \rceil$ et $x_i \leq \lfloor x_i^{PL} \rfloor$.

Outline

- 1 Problème d'optimisation
- 2 Problème d'optimisation linéaire
- 3 Problème d'optimisation linéaire en variables mixtes
- 4 Support Vector Machine

Définition

- Point de vue Optimisation linéaire en nombres entiers (cadre du projet)
- Issu du cours de Fabio Furini
<https://sites.google.com/view/fabiofurini/home-page>

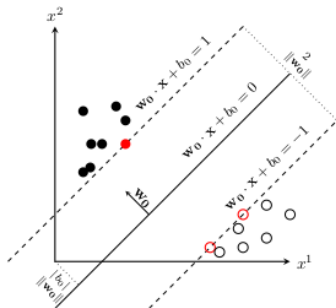
Définiton

- Ensemble de points 2D x_1, \dots, x_n
- Label y_i égal -1 ou 1 pour chaque point

On cherche $w^T x + b = 0$ tel que:

$$y_i(w^T x_i + b) \geq 1 \text{ pour tout } i = 1, \dots, n$$

Ensemble linéairement séparable



On cherche celui qui maximise la marge $\frac{2}{\|w\|}$

Problème Optimisation

$$\min \frac{1}{2} \|w\| \quad (23)$$

$$y_i(w^T x_i + b) \geq 1 \quad \forall i = 1, \dots, n \quad (24)$$

Remarque : On utilisera la norme 1 dans le projet pour une formulation MILP.

Ensemble non linéairement séparable

Deux objectifs

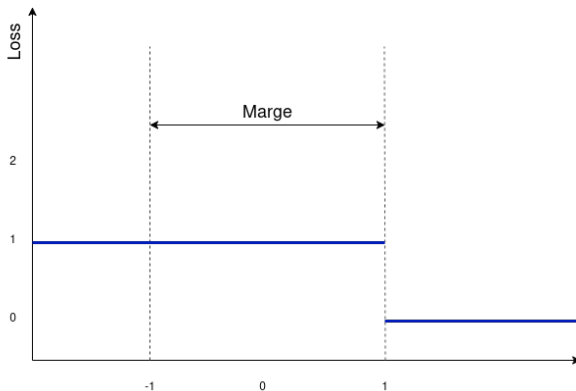
- maximiser la marge
- Minimiser les erreurs de classification

On peut utiliser une des trois fonctions de perte :

- Hard margin loss
- Hinge loss
- Ramp loss

Hard margin loss

Pour chaque point x_i , on a une erreur de 1 si la contrainte $y_i(w^T x_i + b) \geq 1$ n'est pas satisfaite.



Hard margin loss

Ajout d'une variable binaire z_i pour tout $i = 1, \dots, n$ tel que:

$$z_i = \begin{cases} 1 & \text{si } y_i(w^T x_i + b) < 1 \\ 0 & \text{sinon} \end{cases}$$

Formulation

C : poids de la minimisation de l'erreur par rapport à la marge

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n z_i \quad (25)$$

$$y_i(w^T x_i + b) \geq 1 - M z_i \quad \forall i = 1, \dots, n \quad (26)$$

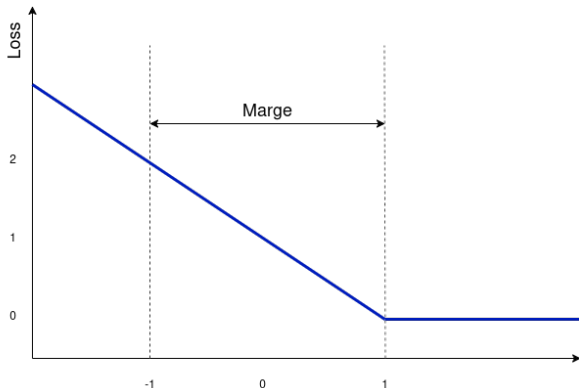
$$z_i \in \{0, 1\} \quad \forall i = 1, \dots, n \quad (27)$$

M : constante suffisamment grande.

Hinge loss

Soit ϵ_i la distance de x_i par rapport à la frontière:

$$\epsilon_i = \begin{cases} 0 & \text{si } y_i(w^T x_i + b) \geq 1 \\ 1 - y_i(w^T x_i + b) & \text{sinon} \end{cases}$$



Hinge loss

Formulation

C : poids de la minimisation de l'erreur par rapport à la marge

$$\min \frac{1}{2} \|w\| + C \sum_{i=1}^n \epsilon_i \quad (28)$$

$$y_i(w^T x_i + b) \geq 1 - \epsilon_i \quad \forall i = 1, \dots, n \quad (29)$$

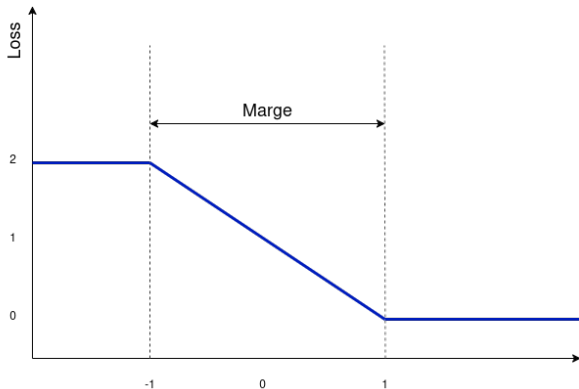
$$\epsilon_i \geq 0 \quad \forall i = 1, \dots, n \quad (30)$$

Ramp loss

Soit ϵ_i l'erreur de x_i si ce dernier est dans la marge :

$$\epsilon_i = \begin{cases} 0 & \text{si } y_i(w^T x_i + b) \geq 1 \text{ ou } y_i(w^T x_i + b) \leq -1 \\ 1 - y_i(w^T x_i + b) & \text{sinon} \end{cases}$$

Si $y_i(w^T x_i + b) \leq -1$ alors on a une erreur de 2 quelle que soit la valeur de $y_i(w^T x_i + b)$.



Ramp loss

Ajout d'une variable binaire z_i pour tout $i = 1, \dots, n$ tel que:

$$z_i = \begin{cases} 1 & \text{si } y_i(w^T x_i + b) \leq -1 \\ 0 & \text{sinon} \end{cases}$$

Formulation

C : poids de la minimisation de l'erreur par rapport à la marge

$$\min \frac{1}{2} \|w\| + C \left(\sum_{i=1}^n \epsilon_i + 2 \sum_{i=1}^n z_i \right) \quad (31)$$

$$y_i(w^T x_i + b) \geq 1 - \epsilon_i - M z_i \quad \forall i = 1, \dots, n \quad (32)$$

$$0 \leq \epsilon_i \leq 2 \quad \forall i = 1, \dots, n \quad (33)$$

$$z_i \in \{0, 1\} \quad \forall i = 1, \dots, n \quad (34)$$

M : constante suffisamment grande.

Linéariser la fonction objective

Minimiser la norme Euclidienne ne peut pas se faire avec l'optimisation linéaire en variables mixtes. On minimise donc la norme 1 (simplification du problème).

Changement de fonction objective

$$\min \frac{1}{2} \|w\|_1 \tag{35}$$

$$y_i(w^T x_i + b) \geq 1 \quad \forall i = 1, \dots, n \tag{36}$$

Remarque : on fait la même chose pour les modèles non linéairement séparables (ie, lorsque l'on ajoute les fonctions de perte précédentes).

Si $w \in \mathbb{R}^k$ ($k = 2$ dans le projet), alors $\|w\|_1 = \sum_{i=1}^k |w_i|$.

Linéariser la fonction objective

Attention : Minimiser une somme de valeurs absolues n'est pas une fonction linéaire !

Linéarisation de la fonction objective

- On ajoute une variable u_j par composante de w .
- On minimise les u_j et on ajoute des contraintes pour que $u_j = |w_j|$ pour toute solution optimale.

On obtient le problème suivant lorsque $w \in \mathbb{R}^k$:

$$\min \frac{1}{2} \sum_{j=1}^k u_j \quad (37)$$

$$u_j \geq w_j \quad \forall j = 1, \dots, k \quad (38)$$

$$u_j \geq -w_j \quad \forall j = 1, \dots, k \quad (39)$$

$$y_i(w^T x_i + b) \geq 1 \quad \forall i = 1, \dots, n \quad (40)$$

Remarque : on fait la même chose pour les modèles non linéairement séparables (ie, lorsque l'on ajoute les fonctions de perte précédentes).