Recherche opérationnelle

2009

Table des matières

Première partie

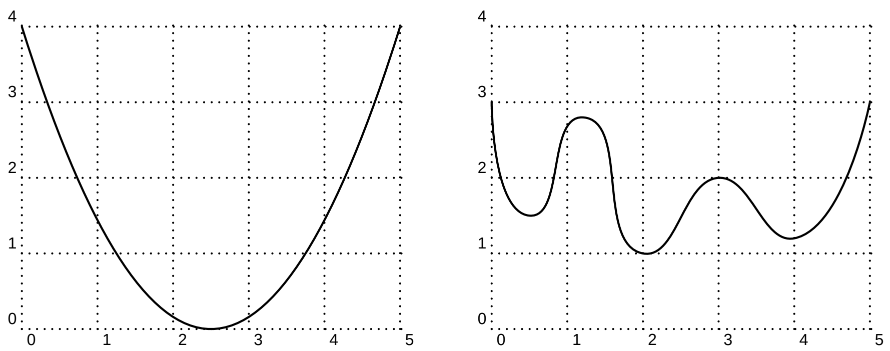
Introduction à la dualité Lagrangienne

Chapitre 1

Introduction

1.1 Introduction

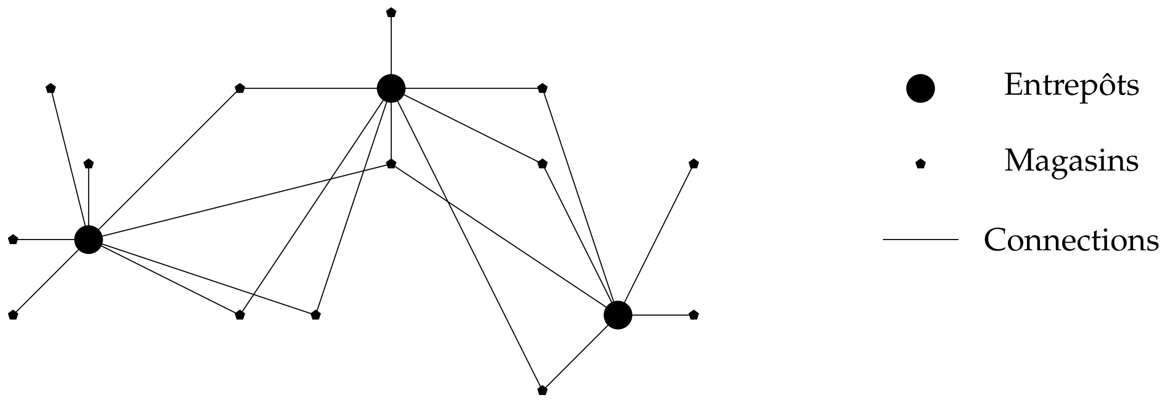
La recherche opérationnelle est l'application industrielle de la théorie de l'optimisation (recherche d'extrêma). Elle a commencé après l'invention de la programmation linéaire par G. Dantzig dans les années 50. Nous rencontrerons dans ce cours des problèmes de minimisation (de coût) ou de maximisation (de profit).



1.2 Quelques exemples

1.2.1 Problème de transport

On considère un problème courant de transport de marchandises. Des magasins (numérotés de 1 à n) ont une demande d_i pour un certain produit. On dispose dans plusieurs entrepôts (numérotés de 1 à m) de stocks s_j. Sachant que le coût unitaire $c_{i,j}$ pour transporter une marchandise d'un entrepôt j à un magasin i est donné, quelle stratégie minimise le coût de transport tout en satisfaisant la demande ?



Si on note $x = (x_{i,j})_{i,j}$ le vecteur des produits transportés de l'entrepôt j au magasin i, le problème est alors de minimiser la fonction

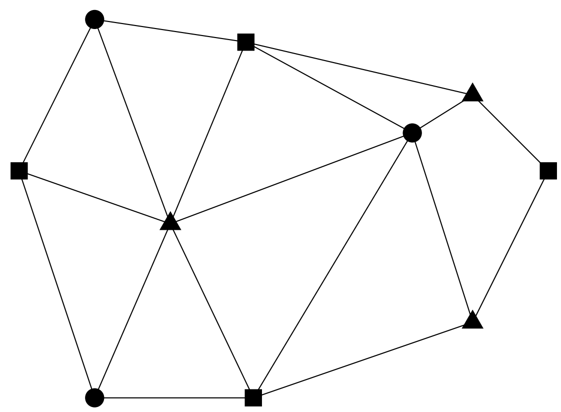
$$f(x) = \sum_{i,j} c_{i,j} x_{i,j}$$

sous les contraintes :

$$\begin{cases} \sum_{j} x_{i,j} = d_i \\ \sum_{i} x_{i,j} \leq s_j \\ x_{i,j} \geq 0 \end{cases}$$

1.2.2 Problème de coloration

On s'interresse au nombre minimum de couleurs nécessaires à la coloration des sommets d'un graphe sous la contrainte que deux sommets reliés (directement) par une arrête ne doivent pas avoir la même couleur.



Existe-t-il une méthode rapide qui trouve sûrement une solution optimale ? La réponse est non. En théorie de la complexité, on introduit plusieurs classes de problèmes. Celui-ci est $\mathcal{NP}$-dur.

Définition 1.2.1. On appelle classe polynômiale (et on note $\mathcal{P}$) la classe des problèmes tels qu'il existe un algorithme résolvant le problème après un nombre d'opérations polynômial en les entrées du problème. On appelle classe non déterministe polynômiale (et non note $\mathcal{NP}$) la classe des problèmes tels qu'il existe un algorithme résolvant le problème avec une probabilité strictement positive après un nombre d'opérations polynômial en les entrées du problèmes.

1.3 Quelques rappels - fondements et notations

Remarque. Maximiser la fonction f revient à minimiser la fonction $-f$. Nous ne nous interresserons donc par la suite qu'aux problèmes du type :

$$\min_{x \in \mathbb{R}^n} f(x), \ x \in C$$

Théorème 1.3.1 (Weirstrass)

Soit C un compact de $\mathbb{R}^n$ et soit $f : C \to \mathbb{R}$ une fonction continue. Alors f atteint son minimum.

Démonstration. On note $F = f(C)$ et $m = Inf(F)$. Par définition de l'inf, il existe une suite $(x_n)_{n\in\mathbb{N}}$ telle que $f(x_n) \underset{n\to+\infty}{\longrightarrow} m$. De plus, C étant compact, on peut extraire de $(x_n)_{n\in\mathbb{N}}$ une sous-suite convergente $(x_{\phi(n)})_{n\in\mathbb{N}}$. Notons x sa limite. Alors x minimise f :

$$m = \lim_{n\to\infty} f(x_n) = \lim_{n\to\infty} f(x_{\phi(n)}) = f(\lim_{n\to\infty} x_{\phi(n)}) = f(x)$$

$\square$

Définition 1.3.2 (gradient)**.** Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction admettant des dérivées partielles. On appelle gradient de f et on note ∇f le vecteur $(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n})^T$.

Définition 1.3.3 (différentielle)**.** Soit $f : \mathbb{R}^n \to \mathbb{R}$. On dit que f est différentiable en $x_0 \in \mathbb{R}^n$ si il existe une application linéaire $L(x_0) \in \mathcal{L}(\mathbb{R}^n, \mathbb{R})$ telle que pour tout $h \in \mathbb{R}^n$:

$$|f(x_0 + h) - f(x_0) - L(x_0)(h)| = o(\|h\|)$$

$L(x_0)$ est la différentielle de f en x_0. On la note en général $df(x_0)$. On peut l'exprimer très simplement en fonction du gradient :

$$df(x_0)(h) = \langle \nabla f(x_0), h \rangle$$

Définition 1.3.4 (hessienne)**.** Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction admettant des dérivées partielles secondes. On appelle hessienne de f la matrice $n \times n$: $\nabla^2 f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{1\leq i,j\leq n}$. Si f est une application deux fois differentiable en x_0 alors on a la formule :

$$d^2 f(x_0)(h) = \langle \nabla^2 f(x_0) h, \cdot \rangle$$

Théorème 1.3.5 (Formule de Taylor)

Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction admettant des dérivées partielles secondes continues en x_0. Alors :

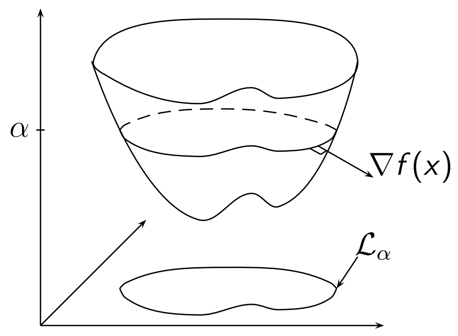
$$\forall h \in \mathbb{R}^n \ : \ f(x+h) = f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2} \langle \nabla^2 f(x) h, h \rangle + o(\|h\|)$$

Définition 1.3.6. Soit $f : \mathbb{R}^n \to \mathbb{R}$. On appelle courbe de niveau α de f l'ensemble $\mathcal{L}_\alpha = \{x \in \mathbb{R}^n : f(x) = \alpha\}$.

Proposition 1.3.7

Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction différentiable. Alors $\nabla f(x)$ est orthogonal à $\mathcal{L}_{f(x)}$.

Démonstration.



Soit d un vecteur unitaire tangent à $\mathcal{L}_{f(x)}$ en x. Alors, il existe une suite $(y_n)_{n \geq 1}$ d'éléments de $\mathcal{L}_{f(x)}$ convergeant vers x telle que :

$$d = \lim_{n \to +\infty} \frac{y_n - x}{\|y_n - x\|}$$

De plus, en écrivant la formule de Taylor à l'ordre 1, on obtient :

$$f(y_n) = f(x) + \langle \nabla f(x), y_n - x \rangle + o(\|h\|)$$

En remarquant que $f(y_n) = f(x)$, puis en divisant par $\|y_n - x\|$ et en passant à la limite lorsque n tend vers l'infini, on obtient : $\langle \nabla f(x), d \rangle = 0$. $\square$

1.4 Exercices

Exercice 0. Soit $S_n(\mathbb{R})$ l'ensemble des matrices symétriques réelles $n \times n$ et soit $A \in S_n(\mathbb{R})$. On rappelle que A est diagonalisable. Montrer que A est diagonalisable dans une base orthogonale de vecteurs propres.

Exercice 1. Soit $a \in \mathbb{R}^n$ et soit f la fonction définie par :

$$\begin{array}{cccc} f & : & \mathbb{R}^n & \to & \mathbb{R} \\ & & x & \mapsto & a^T x \end{array}$$

1. Calculer ∇f
2. Montrer que f est différentiable et calculer df
3. Vérifier la relation $df(x)(h) =< \nabla f(x), h >$

Exercice 2. Soit $A \in M_n(\mathbb{R})$. Montrer que la fonction f définie ci-dessous est différentiable et calculer sa différentielle :

$$\begin{array}{cccc} f & : & \mathbb{R}^n & \to & \mathbb{R} \\ & & x & \mapsto & x^T Ax \end{array}$$

Exercice 3. Soit $S_2(\mathbb{R})$ l'ensemble des matrices symétriques réelles 2×2 et soit $A \in S_2(\mathbb{R})$. On pose :

$$\begin{array}{cccc} f & : & \mathbb{R}^2 & \to & \mathbb{R} \\ & & x & \mapsto & x^T Ax \end{array}$$

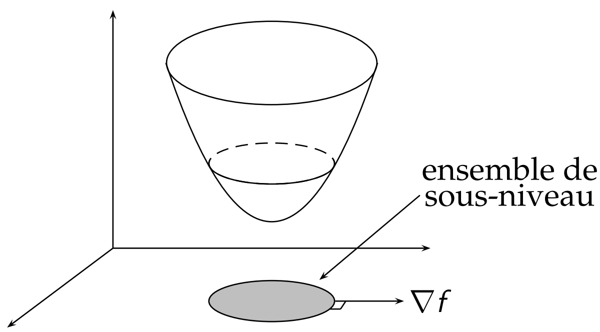
1. En utilisant l'exercice 0, donner une expression simple de f dans une base convenable.
2. Discuter suivant les valeurs propres de A la forme des lignes de niveau.

Chapitre 2

Ensembles convexes, cône tangent et fonctions convexes

2.1 Introduction

Bla bla bla. Bla. Bla bla bla ? Bla ! Bla bla bla bla bla...



2.2 Ensembles convexes, cône tangent et cône normal

2.2.1 Généralités

Définition 2.2.1 (Ensemble convexe). On dit que $C \subset \mathbb{R}^n$ est un ensemble convexe si dès qu'il contient deux éléments, il contient le segment qui les relie :

$$\forall (x, y) \in C^2,\ \forall t \in [0, 1] : tx + (1 - t)y \in C$$

Proposition 2.2.2

Soit I un ensemble d'indices et $(C_i)_{i\in I}$ une famille de convexes. Alors, $\bigcap\limits_{i\in I} C_i$ est un convexe.

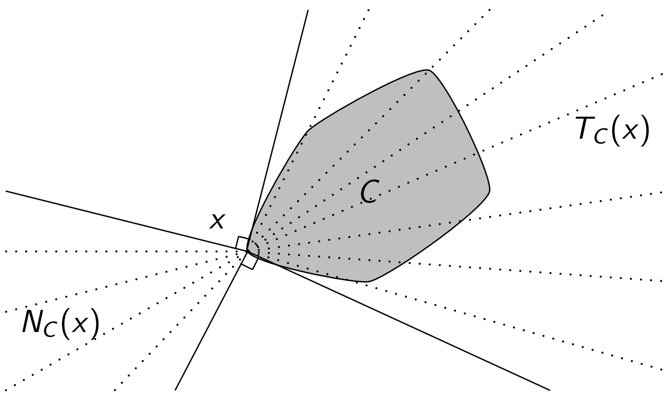
Définition 2.2.3 (Cône tangent). Soit C un convexe et $x \in C$. Le cône tangent à C en x est le plus petit cône fermé contenant $C - \{x\}$. C'est aussi la fermeture de $\{\lambda(y-x), y \in C, \lambda \in \mathbb{R}_+\}$. On le note $T_C(x)$.

Rappel. K est un cône si et seulement si $\forall x \in K,\ \forall \lambda > 0 : \lambda x \in K$.

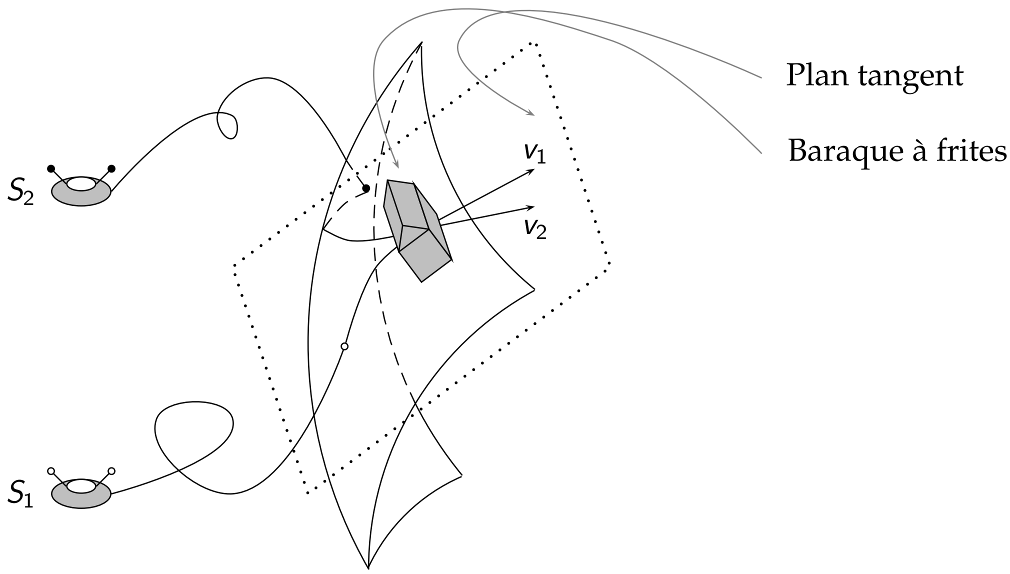
Définition 2.2.4 (Cône Polaire). Soit K un cône de $\mathbb{R}^n$. Le cône polaire à K est $K^\circ = \{v \in \mathbb{R}^n : \forall u \in K, \langle v, u\rangle \leq 0\}$.

Définition 2.2.5 (Cône Normal). Soit C un convexe de $\mathbb{R}^n$ et $x \in C$. Le cône normal à C en x est $N_C(x) = T_C(x)^\circ$.

Remarque. En pratique, on trace les "tangentes" à C en x et on en déduit la forme du cône tangent. Puis, on trace les perpendiculaires aux extrêmités du cône tangent pour en déduire le cône normal.



Remarque. Un extraterrestre arrive en soucoupe volante S_1 sur notre bonne vieille planète terre (dont la forme décrite par toutes les théories modernes de physique est représentée dans le dessin ci-dessous). Ayant très faim, il va au plus court vers le premier magasin de frites F qu'il voit. Comme c'est un extraterrestre, il est clair qu'il peut passer à travers les murs. On note v_1 son vecteur vitesse à l'instant où il arrive au milieu de la baraque à frites. De même, un extraterrestre se pose en soucoupe S_2 sur la terre et va lui aussi au magasin de frites. Soit v_2 son vecteur vitesse à l'instant où il arrive au milieu du magasin. L'ensemble des vecteurs vitesse possibles forme un plan. C'est le plan tangent à la surface en F. Il paraît naturel de dessiner les espaces tangents collés aux surface que l'on considère malgré le fait que les plans tangents sont en fait des espaces vectoriels et devraient plutôt contenir 0. On fait la même chose pour les cônes tangents et normaux qui sont "pointés" en x au lieu de zéro. Cela n'a pas d'incidence puisque nous dessinerons aussi le gradient de la fonction à minimiser en x plutôt qu'en zéro.



Remarque. Pourquoi a-t-on fréquemment des cônes plutôt que des demi-espaces dans la pratique ? Parce que l'ensemble des contraintes est de la forme $C = \{x \in \mathbb{R}^n : g_1(x) \leq 0, \cdots, g_m(x) \leq 0\}$ et qu'il a des "coins" qui correspondent aux x tels que $g_i(x) = g_j(x)$ pour certain i, j.

2.3 Fonctions convexes

Définition 2.3.1 (Epigraphe). Soit $f : \mathbb{R}^n \to \mathbb{R}$. On appelle épigraphe de f l'ensemble $Epi(f) = \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : y \geq f(x)\}$.

Définition 2.3.2 (Fonction convexe). Soit $f : \mathbb{R}^n \to \mathbb{R}$. On dit que f est convexe si son épigraphe est convexe.

Propriété 2.3.3

Soit $f : \mathbb{R}^n \to \mathbb{R}$. Alors, f est convexe si et seulement si :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \forall t \in [0, 1] : f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Définition 2.3.4 (Fonction strictement convexe). Soit $f : \mathbb{R}^n \to \mathbb{R}$. On dit que f est strictement convexe si :

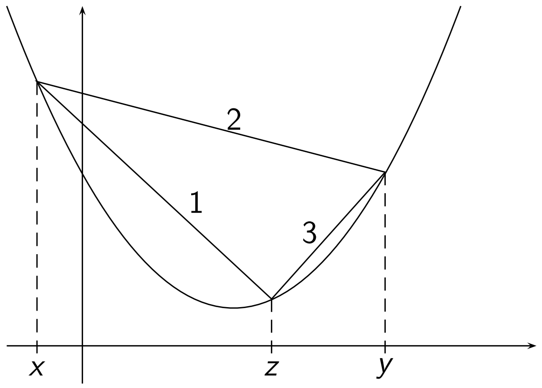
$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, x \neq y, \forall t \in]0, 1[: f(tx + (1-t)y) < tf(x) + (1-t)f(y)$$

2.3.1 Fonctions convexes de $\mathbb{R}$ dans $\mathbb{R}$

Lemme 2.3.5(des trois pentes)

Soit $\varphi : \mathbb{R} \to \mathbb{R}$ une fonction convexe et $x \leq z \leq y$. Alors :

$$\frac{\varphi(z) - \varphi(x)}{z - x} \leq \frac{\varphi(y) - \varphi(x)}{y - x} \leq \frac{\varphi(y) - \varphi(z)}{y - z}$$



Pente 1 $\leq$ Pente 2 $\leq$ Pente 3

Démonstration. Comme $z \in [x, y]$, il existe $t \in [0, 1]$ tel que $z = x + t(y - x)$. Donc, par convexité :

$$\varphi(z) \leq (1 - t)\varphi(x) + t\varphi(y)$$

Soit :

$$\varphi(z) - \varphi(x) \leq t(\varphi(y) - \varphi(x))$$

Mais :

$$t = \frac{z - x}{y - x}$$

On obtient donc finalement :

$$\frac{\varphi(z) - \varphi(x)}{z - x} \leq \frac{\varphi(y) - \varphi(x)}{y - x}$$

L'autre inégalité s'obtient de la même manière. $\square$

Corollaire 2.3.6

Soit $\varphi : \mathbb{R} \to \mathbb{R}$ une fonction convexe dérivable. Alors : φ' croissante $\iff \varphi$ convexe.

Démonstration. $\Leftarrow$: Soit $x \leq z \leq y$. D'après le lemme précédent :

$$\frac{\varphi(z) - \varphi(x)}{z - x} \leq \frac{\varphi(y) - \varphi(x)}{y - x}$$

Donc, en passant à la limite lorsque z tend vers x, on obtient :

$$\varphi^{'}(x) \leq \frac{\varphi(y) - \varphi(x)}{y - x}$$

On a aussi :

$$\frac{\varphi(y) - \varphi(x)}{y - x} \leq \frac{\varphi(y) - \varphi(z)}{y - z}$$

Donc, en passant à la limite lorsque z tend vers y, on obtient :

$$\frac{\varphi(y) - \varphi(x)}{y - x} \leq \varphi^{'}(y)$$

Finalement :

$$\varphi^{'}(x) \leq \frac{\varphi(y) - \varphi(x)}{y - x} \leq \varphi^{'}(y)$$

$\Rightarrow$: Supposons $\varphi^{'}$ croissante. Soit $x \leq z_t \leq y$, $z_t = (1-t)x + ty$. Alors :

$$\varphi(z) - \varphi(x) = \int_x^z \varphi^{'}(u)du \leq \int_x^z \varphi^{'}(z)du = \varphi^{'}(z)(z - x)$$

De même, on a :

$$\varphi(y) - \varphi(z) \geq \varphi^{'}(z)(y - x)$$

Donc :

$$\frac{\varphi(y) - \varphi(z)}{y - z} \geq \varphi^{'}(z) \geq \frac{\varphi(z) - \varphi(x)}{z - x}$$

Ainsi, en multipliant par $y - x$ et en remarquant que $t = (z - x)/(y - x)$, on obtient :

$$\frac{\varphi(y) - \varphi(z)}{1 - t} \geq (y - x)\varphi^{'}(z) \geq \frac{\varphi(z) - \varphi(x)}{t}$$

D'où l'inégalité de convexité recherchée. $\square$

Corollaire 2.3.7

Si φ est $\mathcal{C}^2$, alors : $\varphi^{''} \geq 0 \iff \varphi$ convexe.

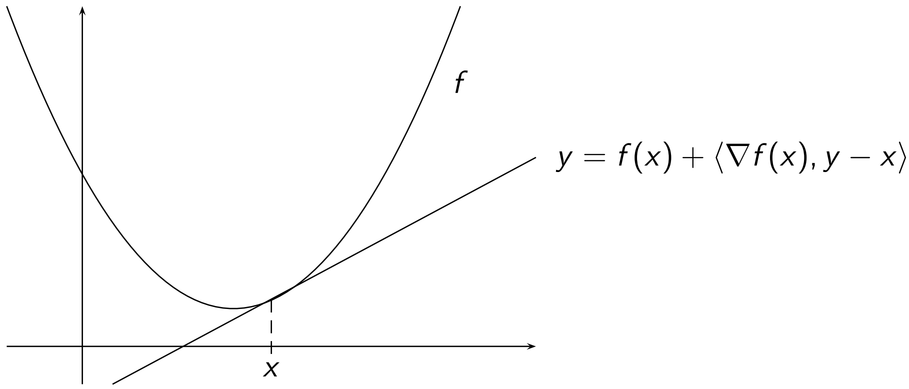
2.3.2 Fonctions convexes de $\mathbb{R}^n$ dans $\mathbb{R}$

Proposition 2.3.8(Inégalité du sous-gradient)

Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe différentiable. Alors :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n : f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

Cette proposition indique qu'une fonction convexe est toujours au-dessus de ses tangentes comme l'illustre la figure ci-dessous :



Démonstration. Soit $x < z_t < y$, où $z_t = x + t(y-x)$. Par convexité, on a :

$$tf(y) + (1-t)f(x) \geq f(z_t)$$

Donc :

$$\begin{aligned} f(y) - f(x) &\geq \frac{f(z_t) - f(x)}{t} \\ &= \frac{f(x + t(y-x)) - f(x)}{t} \\ &= \frac{f(x) + \langle \nabla f(x), t(y-x) \rangle + o(\|y-x\|) - f(x)}{t} \\ &\underset{t \to 0}{\longrightarrow} \langle \nabla f(x), y - x \rangle \end{aligned}$$

□

Définition 2.3.9. Soit $A \in \mathbb{S}_n(\mathbb{R})$. On dit que A est semie-définie positive si pour tout $x \in \mathbb{R}^n$, $\langle Ax, x \rangle \geq 0$. Comme une matrice symétrique réelle est diagonalisable à valeurs propres réelles, cela revient à dire que la plus petite valeur propre de A est positive ou nulle.

Définition 2.3.10. Soit $A \in \mathbb{S}_n(\mathbb{R})$. On dit que A est définie positive si pour tout $x \in \mathbb{R}^n \backslash \{0\}$, $\langle Ax, x \rangle > 0$. C'est équivalent à ce que la plus petite valeur propre de A soit positive.

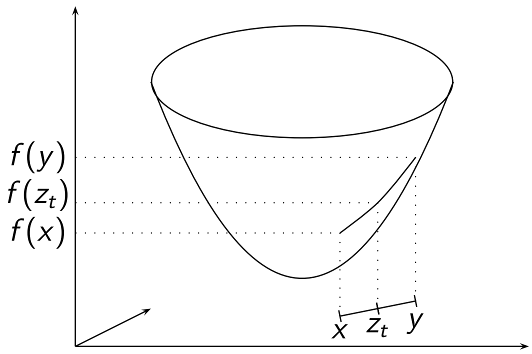
Proposition 2.3.11

Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe de classe $\mathcal{C}^2$ (i.e. deux fois différentiable et dont la différentielle seconde est continue). Si $\nabla^2 f$ est semi-definie positive, alors f est convexe.

Démonstration. Soit $(x, y) \in \mathbb{R} \times \mathbb{R}$ fixé. Pour $t \in [0, 1]$, on note $z_t = tx + (1-t)y$. Pour se ramener au cas scalaire, on définit la fonction :

$$\begin{array}{rccc} \varphi & : & [0,1] & \rightarrow & \mathbb{R} \\ & & t & \mapsto & f(z_t) \end{array}$$

Montrer la convexité de φ (indépendemment de x et y) équivaut à montrer la convexité de f. Nous allons donc montrer que φ est deux fois dérivable et que $\varphi^{''} \geq 0$.



Soit $t \in [0, 1]$ et h petit. Montrons que φ est dérivable en t et donnons l'expression de $\varphi^{'}(t)$:

$$\begin{array}{rcl} \dfrac{\varphi(t+h) - \varphi(t)}{h} & = & \dfrac{f(z_{t+h}) - f(z_t)}{h} \\ & = & \dfrac{\langle \nabla f(z_t), z_{t+h} - z_t \rangle}{h} + \dfrac{o(\|z_{t+h} - z_t\|)}{h} \\ & = & \dfrac{\langle \nabla f(z_t), h(x-y) \rangle}{h} + \dfrac{o(\|h(x-y)\|)}{h} \\ & \underset{h \to 0}{\longrightarrow} & \langle \nabla f(z_t), y - x \rangle = \varphi^{'}(t) \end{array}$$

De même, montrons que $\varphi^{'}$ est dérivable et donnons l'expression de sa dérivée :

$$\begin{array}{rcl} \dfrac{\varphi^{'}(t+h) - \varphi^{'}(t)}{h} & = & \dfrac{\langle \nabla f(z_{t+h}) - \nabla f(z_t), y - x \rangle}{h} \\ & = & \dfrac{\nabla^2 f(z_t)(h)(y-x), y - x}{h} + \dfrac{o(\|h\|)}{h} \\ & \underset{h \to 0}{\longrightarrow} & (y-x)^T \nabla^2 f(z_t)(y-x) = \varphi^{''}(t) \geq 0 \end{array}$$

□

Proposition 2.3.12

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe de classe $\mathcal{C}^2$ (i.e. deux fois différentiable et dont la différentielle seconde est continue). Alors, f est strictement convexe si et seulement si $\nabla^2 f$ est definie positive.

2.4 Conditions d'optimalité

2.4.1 Cas sans contrainte

Théorème 2.4.1

Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe de classe $\mathcal{C}^1$. Alors : $x^* \in \mathbb{R}^n$ minimise $f \iff \nabla f(x^*) = 0$.

Démonstration. $\Rightarrow$: Supposons que x^* minimise f. Alors :

$$\forall y \in \mathbb{R}^n, f(x^*) \leq f(y) = f(x^*) + \langle \nabla f(x^*), y - x^* \rangle + o(\|y - x^*\|)$$

Si $y \neq x$, on peut écrire :

$$\left\langle \nabla f(x), \frac{y - x^*}{\|y - x^*\|} \right\rangle + o(1) \geq 0$$

Donc, en passant à la limite, on obtient que pour tout vecteur unitaire e de $\mathbb{R}^n$, $\langle \nabla f(x^*), e \rangle \geq 0$. Quitte à multiplier par un scalaire, on a :

$$\forall d \in \mathbb{R}^n : \langle \nabla f(x^*), d \rangle \geq 0$$

C'est vrai en particulier pour $-d$ aussi ! Donc, il y a égalité dans l'inégalité précédente. On en déduit que $\nabla f(x^*) = 0$.
$\Leftarrow$: D'après l'inégalité du gradient :

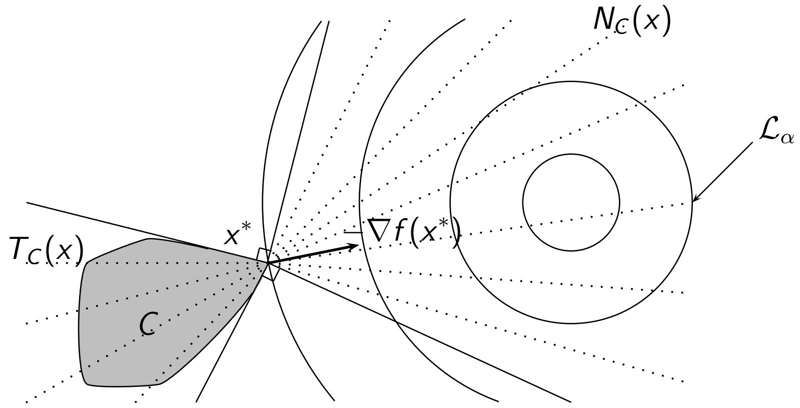
$$\forall y \in \mathbb{R}^n : f(y) \geq f(x^*) + \langle \nabla f(x^*), y - x \rangle = f(x)$$

□

2.4.2 Cas avec contraintes

Théorème 2.4.2

Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction convexe de classe $\mathcal{C}^1$ et $C \subset \mathbb{R}^n$ un convexe fermé. Alors : x^* minimise f sur $C \iff -\nabla f(x^*) \in N_C(x^*)$.



Démonstration. $\Rightarrow$: Supposons que x^* minmise f sur C. Alors :

$$\forall y \in C : f(y) \geq f(x^*)$$

Donc, en utilisant la formule de Taylor :

$$f(x^*) + \langle \nabla f(x^*), y - x^* \rangle + o(\|y - x\|) \geq f(x^*)$$

Ainsi, en divisant par $\|y - x^*\|$ et en passant à la limite, on obtient que pour tout $y \neq x$:

$$\left\langle \nabla f(x^*), \frac{y - x^*}{\|y - x^*\|} \right\rangle \geq 0$$

D'où :

$$\forall \lambda > 0, \forall y \in C : \langle \nabla f(x^*), \lambda(y - x^*) \rangle \geq 0$$

Le résultat est alors prouvé si on se rappelle que $T_C(x^*)$ est la fermeture de $\{\lambda(y - x), y \in C, \lambda \in \mathbb{R}_+\}$ et que $N_C(x) = \{v \in \mathbb{R}^n : \forall u \in K, \langle v, u \rangle \leq 0\}$.

$\Leftarrow$: Supposons que $-\nabla f(x^*) \in N_C(x^*)$. Soit $y \in C$, par l'inégalité du gradient :

$$f(y) \geq f(x) - \langle -\nabla f(x), y - x \rangle \geq f(x)$$

En effet, $y - x \in T_C(x)$ et par hypothèse $-\nabla f(x^*) \in N_C(x^*)$.

$\square$

2.5 Exercices

Exercice 4. Dire si les ensembles suivants sont ouverts, fermés, convexes.

1. $A = \{(x, y, z) \in \mathbb{R}^3 : x + y + z = 0 \text{ et } e^x - y \leq 0\}$
2. $B = \{(x, y) \in \mathbb{R}^2 : y < x < 2y - x^2\}$

Exercice 5. 1. Soit la forme quadratique $Q(x) = x_1^2 + 3x_1x_2 + 5x_2^2$. Cette forme peut s'écrire x^tMx où M est une matrice 2×2. Donnez cette matrice.

2. Soit la matrice M donnée par

$$M = \frac{1}{2}\begin{bmatrix} 0 & a_1 & a_2 \\ a_1 & Q_{11} & Q_{12} \\ a_2 & Q_{12} & Q_{22} \end{bmatrix} \tag{2.1}$$

On pose $x = [x_1 \ \ x_2 \ \ x_3]^t$ et $z = [x_2 \ \ x_3]^t$. Exprimer x^tMx à l'aide de z.

3. Soit la fonction quadratique $f(x) = x^tQx + a^tx$. En se servant de la question précédente, écrire cette fonction comme la restriction d'une forme quadratique de x et d'une variable supplémentaire introduite artificiellement que l'on notera x_0 par exemple. Donner la matrice associée. Cette opération est appelée homogénéisation de la forme quadratique.

Exercice 6. On cherche à minimiser la fonction de $\mathbb{R}^2$ suivante :

$$f(x, y) = -2x + y$$

On suppose de plus que x et y sont soumis aux contraintes :

$$\begin{aligned} 2x + 3y &\geq 6 \\ -x + y &\leq 3 \\ x &\leq 2 \\ x \geq 0 \quad y &\geq 0 \end{aligned}$$

1. Rappeler pourquoi l'ensemble des contraintes est convexe.
2. Représenter sur un dessin les lignes de niveau de f et en déduire la solution du problème.

Exercice 7. 1. Soit $A = \{(x, y) \in \mathbb{R}^2 : x + y \leq 5, y \geq 2 \text{ et } x \geq 2\}$. Décrire le cône tangent à A en $(2, 2)$ et en $(3, 2)$.

2. Soit $B = \{(x, y) \in \mathbb{R}^2 : y \geq x^2 \text{ et } x \geq y^2\}$. Décrire le cône tangent à B en $(0, 0)$ et en $(1/2, 1/4)$.

Exercice 8. Soient f et g deux fonctions de $\mathbb{R}$ dans $\mathbb{R}$.

1. Si chacune est convexe, la somme est-elle convexe ? la différence ? et le produit ?
2. Donner des conditions suffisantes simples pour lesquelles la composée $g \circ f$ est convexe.

Exercice 9. Soit $(f_i)_{i\in I}$ une famille de fonctions convexes sur un espace vectoriel X, indicée par un ensemble quelconque I. Montrer que la fonction f qui à tout x associe la fonction $\sup_{i\in I} f_i(x)$ est convexe.

Exercice 10. On considère l'espace vectoriel $\mathbb{S}_n$ des matrices carrées à coefficients réels et symétriques. Il se trouve que beaucoup de problèmes d'optimisation moderne ont pour variables des matrices et qu'il est donc intéressant de considérer ces matrices comme les variables de base pour notre calcul différentiel. On sait que toute matrice $A \in \mathbb{S}_n$ est diagonalisable et les valeurs propres de A sont réelles.

1. Soit $A \in \mathbb{S}_n$ et soient $\lambda_1 \leq \cdots \leq \lambda_n$ ses valeurs propres et $(e_1, \ldots, e_n)$ une famille orthonormée de vecteurs propres associés. Vérifier que :

$$A = \sum_{k=1}^{n} \lambda_k e_k e_k^t$$

2. Calculer $e_n^t A e_n$.
3. Prenons maintenant n'importe quel vecteur unitaire u. Montrer que $u^t A u \leq \lambda_n$.
4. En déduire que $\lambda_n = \max\limits_{u \text{ t.q. } \|u\|_2 = 1} u^t A u$.
5. On considère maintenant λ_n comme fonction de la matrice A et on veut montrer que c'est une fonction convexe. Montrer que pour un u unitaire donné, la fonction f_u qui à A associe $u^t A u$ est linéaire et donc convexe.
6. Une fonction qui est définie par le maximum sur une famille indicée de fonctions convexes est-elle elle-même convexe ? Si oui, en déduire que la fonction λ_n est convexe.

Exercice 11. Soit $(f_i)_{i \in I}$ une famille de fonctions convexes différentiables sur un espace vectoriel X, indicée par un ensemble **fini** I. On définit :

$$\begin{array}{rccl} f & : & X & \rightarrow & \mathbb{R} \\ & & x & \mapsto & \max\limits_{i \in I} f_i(x) \end{array}$$

Soit $x_0 \in X$ tel que le maximum n'est atteint que pour un seul indice i_0 ($\exists ! i_o \in I$: $f(x_0) = f_{i_0}(x_0)$). Montrer que f est différentiable en x_0 et donner son gradient en x_0. Faire un dessin dans le cas où les f_i sont affines et $X = \mathbb{R}$.

Exercice 12.
1. Soit la fonction $f : \mathbb{R}^2 \mapsto \mathbb{R}$ donnée par $f(x, y) = xy$. Cette fonction est-elle convexe ?
2. On considère n réalisations $Y_1, \ldots, Y_n$ d'une variable aléatoire Y de loi $p_1 \phi(\mu_1, \sigma^2) + p_2 \phi(\mu_2, \sigma^2)$ et on suppose σ^2 connu. Donner la log-vraisemblance des observations. Cette fonction a-t-elle intuitivement une raison de ne pas être concave ?

Chapitre 3

Programmation linéaire, quadratique, et non linéaire

3.1 Programmation linéaire

3.1.1 Définition et premières propriétés

Définition 3.1.1 (programmation linéaire). La programmation linéaire est l'ensemble des problèmes d'optimisation du type :

$$\min_{x\in\mathbb{R}^n} c^T x$$

sous les contraintes :

$$\begin{array}{rcl} a_1^T x & \leq & b_1 \\ & \vdots & \\ a_m^T x & \leq & b_m \end{array}$$

Remarque. On écrit en général le problème de manière compacte : $\min\limits_{x\in\mathbb{R}^n} c^T x$, $Ax \leq b$, où l'inégalité a lieu coordonnée par coordonnée.

Définition 3.1.2 (Forme standard d'un programme linéaire). Un problème de programmation linéaire est dit standard lorsqu'il est de la forme :

$$\min_{x\in\mathbb{R}^n} c^T x$$

sous les contraintes :

$$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

Proposition 3.1.3

Tout problème de programmation linéaire peut se mettre sous forme standard.

Démonstration. Plutôt que de donner une démonstration détaillée de cette proposition, nous en donnerons les grandes lignes puis nous traiterons un exemple. Soit donné un problème de programmation linéaire :

$$\min_{x\in\mathbb{R}^n} c^T x,\ Ax \leq b$$

Pour ajouter la contrainte $x \geq 0$, on écrit simplement $x = x^+ - x^-$. Comment transformer alors l'inégalité en égalité ? En ajoutant des variables ! Posons :

$$\begin{cases} y = b - Ax \geq 0 \\ X = (x\ \ y) \\ \overline{A} = (A\ \ Id) \end{cases}$$

où Id est la matrice identité de la bonne taille. Le problème est devenu :

$$\min_{x\in\mathbb{R}^n} c^T x,\ \overline{A}X = b$$

□

Exemple. On considère le problème de programmation linéaire suivant :

$$\min_{x\in\mathbb{R}^3} 3x_1 - x_3$$

sous les contraintes :

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 - x_2 - x_3 \leq 1 \\ x_1 + x_3 \geq -1 \\ x_1 \geq 0,\ x_2 \geq 0 \end{cases}$$

Comme il n'y a pas de restriction de signe sur x_3, on pose : $x_3 = y_1 - y_2$, où $y_1 = x_3^+ \geq 0$ et $y_2 = x_3^- \geq 0$. En multipliant la deuxième inégalité par -1 et en notant $x^{'} = (x_1, x_2, y_1, y_2)^T$, le problème devient :

$$\min_{x^{'}\in\mathbb{R}^4} 3x_1 - y_1 + y_2$$

sous les contraintes :

$$\begin{cases} x_1 + x_2 + y_1 - y_2 = 1 \\ x_1 - x_2 - y_1 + y_2 \leq 1 \\ -x_1 - y_1 + y_2 \leq 1 \\ x_1 \geq 0,\ x_2 \geq 0,\ y_1 \geq 0,\ y_2 \geq 0 \end{cases}$$

Enfin, pour transformer les inégalités en égalités, on introduit encore deux nouvelles variables positives z_1 et z_2. Si on note $x^{''} = (x_1, x_2, y_1, y_2, z_1, z_2)^T$ problème devient :

$$\min_{x^{''}\in\mathbb{R}^6} 3x_1 - y_1 + y_2$$

sous les contraintes :

$$\begin{cases} x_1 + x_2 + y_1 - y_2 = 1 \\ x_1 - x_2 - y_1 + y_2 + z_1 = 1 \\ -x_1 - y_1 + y_2 + z_2 = 1 \\ x_1 \geq 0,\ x_2 \geq 0,\ y_1 \geq 0,\ y_2 \geq 0,\ z_1 \geq 0,\ z_2 \geq 0 \end{cases}$$

3.1.2 Cône normal à un polyèdre

Pour résoudre les problèmes de programmation linéaire, on est amené à calculer le cône normal à un polyèdre $C = \{x \in \mathbb{R}^n : Ax \leq b\}$ en un point x de ce polyèdre. Nous donnons dans ce paragraphe une expression simple du cône normal à un polyèdre. Dans tout le paragraphe nous considèrerons un polyèdre de la forme $C = \{x \in \mathbb{R}^n : Ax \leq b\}$ où l'inégalité a lieu coordonnée par coordonnée. On note a_j le j-ième vecteur ligne de la matrice A de sorte que l'on a : $x \in C \iff \forall j \in [\![1, m]\!] : a_j^T x \leq b_j$. Avant d'énoncer le théorème, nous donnons une version simplifiée du théorème de Hahn-Banach dont nous aurons besoin lors de la démonstration.

Théorème 3.1.4 (Hahn-Banach)

Soit M un cône convexe fermé de $\mathbb{R}^n$ et $g \notin M$. Alors, il existe un hyperplan séparant g et M qui passe par l'origine. Autrement dit :

$$\exists u \in \mathbb{R}^n \ / \ \begin{cases} \forall z \in M, \langle u, z \rangle \leq 0 \\ \langle u, g \rangle > 0 \end{cases}$$

Définition 3.1.5 (Contrainte active). Une contrainte $a_j^T x \leq b_j$ sera dite active en x si $a_j^T x = b_j$. On note $\mathcal{J}_x$ l'ensemble des indices des contraintes actives en x.

Théorème 3.1.6 (Caractérisation du cône normal pour les polyèdres)

Soit $C = \{x \in \mathbb{R}^n : Ax \leq b\}$ un polyèdre et $x \in C$. Alors, $N_C(x) = \{\sum_{j \in \mathcal{J}_x} \lambda_j a_j, \lambda_j \geq 0\}$.

Démonstration. $\supset$: Soit $g = \sum_{j \in \mathcal{J}_x} \lambda_j a_j$ et soit $y \in C$. Montrons que $\langle g, y - x \rangle \leq 0$. On sait que :

$$\begin{aligned} \forall j \in \mathcal{J}_x : \quad & a_j^T x = b_j \\ \forall j \in [\![1, m]\!] : \quad & a_j^T y \leq b_j \end{aligned}$$

Donc :

$$\forall j \in \mathcal{J}_x : a_j^T (y - x) \leq 0$$

D'où :

$$\forall j \in \mathcal{J}_x : \lambda_j a_j^T (y - x) \leq 0$$

Finalement,

$$\langle g, y - x \rangle = \langle \sum_{j \in \mathcal{J}_x} \lambda_j a_j, y - x \rangle = \sum_{j \in \mathcal{J}_x} \lambda_j a_j^T (y - x) \leq 0$$

Le résultat est encore vrai en remplaçant $y - x$ par $v \in T_C(x)$ par stabilité de l'inégalité précédente par multiplication par un scalaire positif et par passage à la limite. Donc, $g \in N_C(x)$.

$\subset$: Notons $M = \{\sum_{j \in \mathcal{J}_x} \lambda_j a_j, \lambda_j \geq 0\}$. Soit $g \notin M$. Montrons que $g \notin N_C(x)$. D'après le théorème de Hahn-Banach :

$$\exists u \in \mathbb{R}^n \ / \ \begin{cases} \forall z \in M, \langle u, z \rangle \leq 0 \\ \langle u, g \rangle > 0 \end{cases}$$

Posons $y = x + \varepsilon u$ et montrons que l'on peut choisir $\varepsilon > 0$ suffisamment petit pour que y soit un élément de C. On sait que si $j \in \mathcal{J}_x$, $a_j^T u \leq 0$ car $a_j \in M$. Donc :

$$\forall j \in \mathcal{J}_x \ : \ a_j^T y = a_j^T(x + \varepsilon u) = a_j^T x + \varepsilon a_j^T u \leq b_j + \varepsilon a_j^T u \leq b_j$$

Si maintenant $j \notin \mathcal{J}_x$, on sait que $a_j^T x < b_j$, et donc, on peut choisir ε tel que $a_j^T x + \varepsilon a_j^T u \leq b_j$, et ce, indépendemment de j. Donc :

$$\forall j \notin \mathcal{J}_x \ : \ a_j^T y = a_j^T(x + \varepsilon u) = a_j^T x + \varepsilon a_j^T u \leq b_j$$

Finalement, y est bien un élément de C. On conclut en remarquand que :

$$\langle g, y - x \rangle = \varepsilon \langle g, u \rangle > 0$$

Ainsi, $g \notin N_C(x)$. $\square$

3.2 Programmation quadratique

3.2.1 Définition et équations de KKT

Définition 3.2.1 (Programmation quadratique). La programmation quadratique est l'ensemble des problèmes d'optimisation du type :

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + q^T x$$

sous les contraintes :

$$Ax \leq b$$

où $Q \in S_n(\mathbb{R})$ et $q \in \mathbb{R}^n$.

Remarque. On impose dans cette définition que la matrice Q soit symétrique. Dans le cas général, on peut simplement se ramener à une matrice symétrique. En effet, soit $A \in M_n(\mathbb{R})$, alors, comme $x^T A x = (x^T A x)^T = x^T A^T x$, on a :

$$x^T \left(\frac{A + A^T}{2} \right) x = x^T A x$$

et la matrice $(A + A^T)/2$ est symétrique.

Remarquons aussi que la fonction à maximiser étant de classe $\mathcal{C}^2$, elle est convexe si et seulement si sa hessienne est définie positive c'est-à-dire si et seulement si Q est définie positive.

3.2.2 Le modèle de Markovitz

On considère n actions dont les prix d'achat à l'instant t sont $(S_1(t), \cdots, S_n(t))$. On dispose d'un budget B que l'on veut investir dans ces actions. Notons x_i le nombre d'actions i que l'on veut acheter. On veut maximiser le profit. On note f le profit.

$$S(t) = (S_1(t), \cdots, S_n(t))^T, \ \rho = S(t+1) - S(t), \ f(x) = \sum_{i=1}^{n} x_i(S_i(t+1) + S_i(t)) = \rho^T x$$

On cherche donc à résoudre :

$$\max_{x \in \mathbb{R}^n} \ \rho^T x, \ S^T x \leq B, \ x \geq 0$$

Le problème est qu'on ne connaît pas $S(t+1)$! Mais, on peut estimer la loi des accroissements $S(t+1) - S(t)$ en se basant par exemple sur le prix des actions au cours des 15 derniers jours. Une approche naturelle mais naïve consiste alors à tenter de maximiser l'espérance du profit : $\mathbb{E}(f(x)) = \mathbb{E}(\rho^T x) = \mathbb{E}(\rho)^T x$. Cette approche est en fait très risquée puisque la solution consiste à tout investir dans l'action dont l'espérance du taux d'accroissement est la plus grande. Markovitz a introduit la notion de risque. Mathématiquement, le risque est lié à la variance. Plus la variance est élevée, plus le risque est élevé. On définit le risque par :

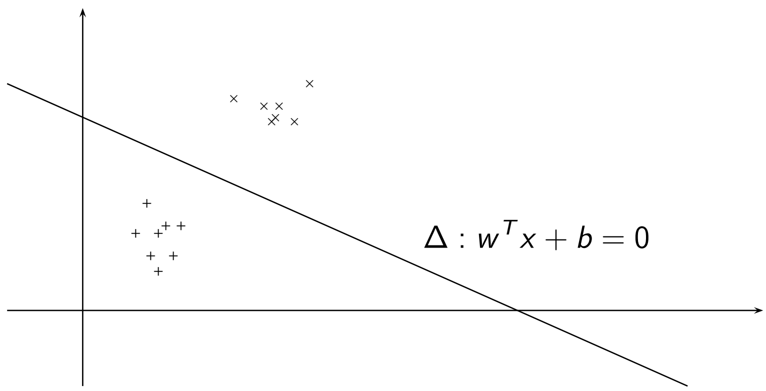
$$\begin{aligned}
Var \rho^T x &= \mathbb{E}((\rho^T x - \mathbb{E}(\rho)^T x)^2) \\
&= \mathbb{E}((\rho^T x - \mathbb{E}(\rho)^T x)^T (\rho^T x - \mathbb{E}(\rho)^T x)) \\
&= \mathbb{E}(x^T (\rho - \mathbb{E}(\rho))(\rho - \mathbb{E}(\rho))^T x) \\
&= x^T \mathbb{E}((\rho - \mathbb{E}(\rho))(\rho - \mathbb{E}(\rho)^T))x \\
&= x^T Q x
\end{aligned}$$

où Q est la matrice de covariance des accroissements. On se fixe un rendement r. Le problème de Markovitz est alors :

$$\min_{x \in \mathbb{R}^n} x^T Q x, \quad \mathbb{E}(\rho)^T x \geq r, \ S^T x \leq B, \ x \geq 0$$

3.2.3 Le problème du learning

On dispose d'un ensemble $(x_1, \cdots x_n)$ de points que l'on peut séparer en deux sous-groupes et on cherche la meilleure droite de séparation. Comment se ramener à un problème quadratique ?



Le problème est donc de maximiser $\min d(x_i, \Delta)$, où Δ est la droite d'équation $w^T x + b = 0$. Les paramètres sont w et b.

Lemme 3.2.2

La distance d'un point x à la droite Δ s'exprime par la formule :

$$d(x, \Delta) = \frac{|w^T x + b|}{\|w\|}$$

Démonstration. Soit $\rho_\Delta(x)$ la projection de x sur Δ. Comme w est un vecteur normal à Δ, on sait qu'il existe un réel λ tel que $\rho_\Delta(x) = x + \lambda w$. De plus, comme $\rho_\Delta(x) \in \Delta$, on a aussi : $w^T \rho_\Delta(x) + b = 0$. On en déduit successivement que :

$$w^T(x + \lambda w) + b = 0$$

puis que :

$$\lambda = \frac{-w^T x - b}{\|w\|^2}$$

Finalement :

$$d(x, \Delta) = \|x - \rho_\Delta(x)\| = \|\lambda w\| = \frac{|w^T x + b|}{\|w\|}$$

$\square$

Il faut bien sûr prendre en compte le fait que la droite Δ doit séparer les deux sous-groupes ! On introduit pour cela des variables $(y_1, \cdots y_n)$. On affecte alors à la variable y_i la valeur 1 si x_i appartient au premier sous-groupe, et la valeur -1 si x_i appartient au deuxième sous-groupe. On peut traduire cela simplement par :

$$\begin{aligned} y_i = 1 &\iff w^T x_i + b > 0 \\ y_i = -1 &\iff w^T x_i + b < 0 \end{aligned}$$

De plus, comme les x_i sont en nombre fini, on sait que le maximum sur (w, b) de $\min |w^T x + b|$ est strictement positif (on peut toujours trouver une droite séparante qui ne touche aucun point, à condition que le problème admette une solution). Donc, quitte à multiplier w et b par un scalaire, on peut imposer : $\min |w^T x + b| = 1$. En effet, si on a trouvé une solution (w, b) au problème telle que $\min |w^T x + b| = \alpha > 0$, alors $(w/\alpha, b/\alpha)$ est aussi solution et vérifie l'égalité. Le problème de départ est donc équivalent à :

$$(P) \qquad \min_{(w,b)} \|w\|^2, \quad \begin{cases} y_i(w^T x_i + b) & \geq & 1 \\ \min |w^T x_i + b| & = & 1 \end{cases}$$

Lemme 3.2.3

Le problème (P) est en fait équivalent à :

$$(\widetilde{P}) \qquad \min_{(w,b)} \|w\|^2, \quad y_i(w^T x_i + b) \geq 1$$

Démonstration. • Soit $i \in [\![1, n]\!]$, alors :

$$|w^T x_i + b| = |y_i(w^T x_i + b)| \geq 1$$

Donc, $\min |w^T x_i + b| \geq 1$

• Soit (w, b) une solution de $(\widetilde{P})$. Supposons que $\min |w^T x_i + b| > 1$. Soit i_0 un indice où ce minimum est atteint. On pose :

$$\widehat{w} = \frac{w}{|w^t x_{i_0} + b|} \ , \ \ \widehat{b} = \frac{b}{|w^t x_{i_0} + b|}$$

Alors, $(\widehat{w}, \widehat{b})$ est une meilleur solution que (w, b) ce qui est absurde. En effet, par construction : $\|\widehat{w}\|^2 < \|w\|^2$. De plus :

$$\forall i \in [\![1, n]\!], \ \ |\widehat{w}^T x_i + \widehat{b}| = \frac{|w^t x_i + b|}{|w^t x_{i_0} + b|} = \frac{|w^t x_i + b|}{|w^t x_i + b|} = 1$$

□

Proposition 3.2.4

$(\widetilde{P})$ est un problème de programmation quadratique.

Démonstration. Posons :

$$A = \begin{pmatrix} -x_i^T - 1 \\ \vdots \\ x_i^T + 1 \end{pmatrix} \ , \ \ Q = \begin{pmatrix} 2I & 0 \\ 0 & 0 \end{pmatrix} \ , \ \ q = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \ , \ \ b = \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix}$$

Alors, le problème $(\widetilde{P})$ s'écrit :

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + q^T x \ , \quad Ax \leq b$$

□

3.3 Programmation non linéaire convexe

Définition 3.3.1 (Programmation non linéaire). La programmation non linéaire convexe est l'ensemble des problèmes d'optimisation du type :

$$\min_{x\in\mathbb{R}^n} f(x)$$

sous les contraintes :

$$g_1(x)=0,\cdots,g_{n_e}(x)=0, g_{n_e+1}(x)\leq 0,\cdots,g_{n_e+n_i}(x)\leq 0$$

où les fonctions $g_1,\cdots g_{n_e}$ sont affines et les fonctions $g_{n_e+1},\cdots g_{n_e+n_i}$ sont convexes.

Remarque. On impose dans la définition que les contraintes d'égalité soient réalisées pour des fonctions affines de sorte que les ensembles $\{x\in\mathbb{R}^n : g_j(x)=0\}$ soient convexes. L'ensemble de niveau 0 d'une fonction peut être convexe sans que la fonction soit affine mais en pratique c'est à des fonctions affines que nous rencontrerons.

Théorème 3.3.2

Soit C l'ensemble admissible défini par les contraintes de la définition, et soit $x\in C$. On suppose qu'il existe $\bar{u}\in T_C(x)$ tel que :

$$\forall j\in\mathcal{J}_x\ :\ \langle\nabla g_j(x),\bar{u}\rangle<0$$

On suppose de plus que la famille $(\nabla g_1(x),\cdots,\nabla g_{n_e}(x))$ est libre. Alors, le cône tangent à C en x est :

$$T_C(x)=\{u\in\mathbb{R}^n,\forall j\in\mathcal{J}_x,\langle\nabla g_j(x),u\rangle\leq 0\ et\ \forall k\in[\![1,n_e]\!],\langle u,\nabla g_k(x)\rangle=0\}$$

Démonstration dans le cas sans contrainte d'égalité :
$\subset$: Soit $u\in T_C(x)$. Soit $y\in C$, on pose : $z_t=x+t(y-x)$. Alors, $z_t\in C$. Donc :

$$\forall j\in\mathcal{J}_x,\ g_j(z_t)\leq 0$$

Or,

$$\forall j\in\mathcal{J}_x,\ g_j(x)=0$$

Donc, pour tout t positif :

$$\frac{g_j(z_t)-g_j(x)}{t}\leq 0$$

Par ~~Toutatis~~ Taylor :

$$\begin{aligned}0\leq\ \frac{1}{t}(g_j(z_t)-g_j(x)) &= \frac{1}{t}\left(\langle\nabla g_j(x),z_t-x\rangle+o(\|z_t-x\|)\right)\\ &= \langle\nabla g_j(x),\frac{t(y-x)}{t}\rangle+\frac{o(t)}{t}\\ &\underset{t\to0^+}{\longrightarrow} \langle\nabla g_j(x),y-x\rangle\end{aligned}$$

Donc, pour tout $y \in C$, pour tout $\lambda \geq 0$, on a : $\langle \nabla g_j(x), \lambda(y-x) \rangle$. D'où le résultat par passage à la limite.
$\supset$: Soit $u \in \mathbb{R}^n$ tel que $\forall j \in \mathcal{J}_x$, $\langle \nabla g_j(x), u \rangle \leq 0$.

– Montrons que $\bar{u} \in T_C(x)$. On a :

$$\forall j \notin \mathcal{J}_x, g_j(x) < 0$$

Donc, pour $j \notin \mathcal{J}_x$ et t_j petit, $g_j(x + t\bar{u}) < 0$. De plus :

$$\forall j \in \mathcal{J}_x,\ g_j(x+t\bar{u})-g_j(x) = \nabla g_j(x), x+t\bar{u}-x\rangle + o(\|x+t\bar{u}-x\|) = t(\langle \nabla g_j(x), \bar{u}\rangle + o(t)/t)$$

Donc, pour $j \in \mathcal{J}_x$ et t_j petit, $g_j(x+t\bar{u}) < 0$. Ainsi, en prenant t suffisament petit (par exemple $t = \min t_j$), $x + t\bar{u} \in C$. D'où :

$$\bar{u} = \frac{1}{t}((x + t\bar{u}) - x) \in T_C(x)$$

– Posons $u_t = u + t\bar{u}$. Alors :

$$\forall j \in \mathcal{J}_x,\ \langle \nabla g_j(x), u_t \rangle = \langle \nabla g_j(x), u \rangle + t\langle \nabla g_j(x), \bar{u} \rangle < 0$$

Donc, $u_t \in T_C(x)$ pour la même raison que $\bar{u} \in T_C(x)$. De plus, $T_C(x)$ est fermé, donc :

$$u = \lim_{t \to 0^+} u_t \in T_C(x)$$

$\square$

Corollaire 3.3.3

Soit C l'ensemble admissible défini par les contraintes de la définition, et soit $x \in C$. On suppose qu'il existe $\bar{u} \in C$ tel que :

$$\forall j \in \mathcal{J}_x \ : \ \langle \nabla g_j(x), \bar{u} \rangle < 0$$

On suppose de plus que la famille $(\nabla g_1(x), \cdots, \nabla g_{n_e}(x))$ est libre. Alors, le cône normal à C en x est :

$$N_C(x) = \{ \sum_{j \in \mathcal{J}_x \cup [\![1,n_e]\!]} \lambda_j \nabla g_j(x),\ \ \forall j \in \mathcal{J}_x : t_j \geq 0\}$$

3.4 Equations de Karush-Kuhn-Tucker

On considère le problème :

$$\min_{x \in \mathbb{R}^n} f(x)$$

sous les contraintes :

$$g_1(x) = 0, \cdots, g_{n_e}(x) = 0, g_{n_e+1}(x) \leq 0, \cdots, g_{n_e+n_i}(x) \leq 0$$

On note C l'ensemble des points admissibles pour ce problème. On suppose que le cône normal à C en x est donné par :

$$N_C(x) = \{ \sum_{j \in \mathcal{J}_x \cup [\![1,n_e]\!]} \lambda_j \nabla g_j(x), \ \ \forall j \in \mathcal{J}_x : t_j \geq 0\}$$

Alors, on a montré que x est optimal si et seulement si :

$$\exists (\lambda_1, \cdots, \lambda_{n_e}, \lambda_{n_e+1}, \cdots \lambda_{n_e+n_i}) \in \mathbb{R}^{n_e} \times \mathbb{R}_+^{n_i} \ : \ \begin{cases} \nabla f(x) + \sum \lambda_j \nabla g_j(x) = 0 \\ \forall j \in \mathcal{J}_x : \lambda_j g_j(x) = 0 \end{cases}$$

Ces équations sont appelées équations de Karush-Kuhn-Tucker.

3.5 Exercices

Exercice 13. On cherche à minimiser la fonction :

$$\begin{array}{rcccc} f & : & \mathbb{R}^2 & \to & \mathbb{R} \\ & & (x,y) & \mapsto & x^2 + y^2 \end{array}$$

sous la contrainte :

$$Max(|x|,|y|) \leq 1$$

1. Ecrire le problème sous forme d'un problème d'optimisation convexe.
2. Résoudre le problème.

Exercice 14. Minimiser la fonction f de $\mathbb{R}^2$ dans $\mathbb{R}$:

$$f(x,y) = -2x + y$$

sous les contraintes :

$$\begin{aligned} 2x + 3y &\geq 6 \\ -x + y &\leq 3 \\ x &\leq 2 \\ x \geq 0 \quad y &\geq 0 \end{aligned}$$

Exercice 15. Minimiser :

$$g(x,y) = 20x + 10y + 3x^2 + 2y^2$$

sous les contraintes :

$$\begin{aligned} 2x + y &\leq 6 \\ x + y &\leq 10 \\ 2x + 3y &\geq 8 \\ x \geq 0 \quad y &\geq 0 \end{aligned}$$

Exercice 16. Maximiser :

$$h(x, y) = 8x + 4y - x^2 - y^2$$

sous les contraintes :

$$\begin{gathered} x + y \leq 2 \\ x \geq 0 \quad y \geq 0 \end{gathered}$$

Chapitre 4

Dualité Lagrangienne

4.1 Introduction

Pour utiliser les équations de Karush-Kuhn-Tucker, il faut proposer un candidat x et vérifier qu'il convient. Il faut donc une méthode simple et systématique : la dualité. Même lorsque le problème est trop compliqué, la dualité va quand même apporter des informations non explicites dans le problème initial.

4.2 Fonction de Lagrange et problème dual dans le cas de contraintes d'égalité

Le problème que l'on considère dans cette section est :

$$\max_{x \in \mathfrak{X}} f(x)$$

sous les contraintes :

$$g_1(x) = 0, \dots, g_m(x) = 0$$

Pour simplifier l'écriture, on notera g le vecteur $(g_1, \dots, g_m)^T$, et C l'ensemble des points admissibles. On suppose de plus que toutes les fonctions du problème sont continues et que $\mathfrak{X}$ est une partie de $\mathbb{R}^n$.

Définition 4.2.1 (Fonction de Lagrange). La fonction de Lagrange associée au problème est :

$$\begin{array}{rcccl} L & : & \mathfrak{X} \times \mathbb{R}^m & \longrightarrow & \mathbb{R} \\ & & (x, \lambda) & \mapsto & f(x) + \langle \lambda, g(x) \rangle \end{array}$$

Remarque. Si on pose :

$$\begin{array}{rcccl} \eta & : & \mathbb{R}^n & \longrightarrow & \mathbb{R} \\ & & x & \mapsto & \inf_{\lambda \in \mathbb{R}^n} L(x, \lambda) = \begin{cases} -\infty & \text{si } \exists j : g_j(x) \neq 0 \\ f(x) & \text{sinon} \end{cases} \end{array}$$

Alors,

$$\sup_{x\in C} f(x) = \sup_{x\in\mathfrak{X}} \eta(x) = \sup_{x\in\mathfrak{X}} \inf_{\lambda\in\mathbb{R}^n} L(x,\lambda)$$

La dualité lagrangienne consiste à regarder le problème où le sup et l'inf sont pris dans l'ordre inverse.

Définition 4.2.2 (Fonction duale). La fonction duale du problème de départ est :

$$\begin{array}{cccc} \theta & : & \mathbb{R}^m & \longrightarrow & \mathbb{R} \\ & & \lambda & \longmapsto & \sup_{x\in\mathfrak{X}} L(x,\lambda) \end{array}$$

Proposition 4.2.3(miraculeuse)

La fonction duale θ est convexe.

Lemme 4.2.4

Soit $(f_i)_{i\in I}$ une famille de fonctions convexes. Alors, la fonction $\sup_{i\in I} f_i$ est convexe.

Démonstration du lemme 4.2.4. Commençons par remarquer que l'épigraphe du sup est égal à l'intersection des épigraphes. En effet :

$$(x,y) \in \bigcap_{i\in I} epi(f_i) \Leftrightarrow \forall i \in I : y \geq f(x) \Leftrightarrow y \geq \sup_{i\in I} f_i(x) \Leftrightarrow (x,y) \in epi(\sup_{i\in I} f_i))$$

On sait de plus qu'une intersection de convexes est convexe. Donc, l'épigraphe de la fonction $\sup_{i\in I} f_i$ étant convexe, le résultat est démontré. □

Démonstration de la proposition miraculeuse. A x fixé, la fonction $\lambda \mapsto L(x,\lambda)$ est linéaire en u, donc convexe. Ainsi, la fonction $\theta = \sup_{x\in\mathfrak{X}} L(x,\cdot\,)$ est convexe d'après le lemme. □

Théorème 4.2.5 (dualité faible)

On a :

$$\inf_{\lambda\in\mathbb{R}^n} \theta(\lambda) \geq \sup_{\substack{g(x)=0 \\ x\in\mathfrak{X}}} f(x)$$

Démonstration. Par définition du lagrangien :

$$\theta(\lambda) = \sup_{x\in\mathfrak{X}} L(x,\lambda) \geq \sup_{\substack{g(x)=0 \\ x\in\mathfrak{X}}} L(x,\lambda) = \sup_{\substack{g(x)=0 \\ x\in\mathfrak{X}}} (f(x) + \langle\lambda, g(x)\rangle) = \sup_{\substack{g(x)=0 \\ x\in\mathfrak{X}}} f(x)$$

D'où le résultat en passant à l'inf dans l'inégalité précédente. □

Définition 4.2.6 (sous-gradient)**.** Soit θ une fonction convexe et λ tel que $\theta(\lambda) < \infty$. On dit que g est un sous-gradient de θ en λ si :

$$\forall \mu, \quad \theta(\mu) \geq \theta(\lambda) + \langle \mu - \lambda, g \rangle$$

Autrement dit, l'hyperplan dirigé par g passant par $\theta(\lambda)$ est en-dessous de l'épigraphe de θ.

Définition 4.2.7 (sous-différentiel)**.** Soit θ une fonction convexe et λ tel que $\theta(\lambda) < \infty$. Le sous différentiel de θ en λ est l'ensemble des sous-gradients de θ et λ. On le note $\partial\theta$.

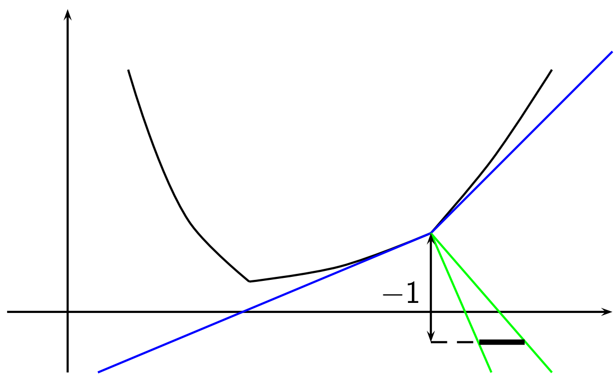


FIG. 4.1 – Sous-gradient et sous-différentiel

Remarque. $\partial\theta(\lambda)$ est un convexe fermé. De plus, on a la condition d'optimalité suivante :

$$0 \in \partial\theta(\lambda) \Rightarrow \lambda \text{ optimal}$$

Proposition 4.2.8

Pour tout $\lambda \in \mathbb{R}^m$, on définit $\mathfrak{X}_\lambda = \{x \in \mathbb{R}^n : x \text{ maximise } L(\cdot\,, \lambda)\}$. Alors :

$$\overline{\text{Conv}}\; g(\mathfrak{X}_\lambda) \subset \partial\theta(\lambda)$$

où $\overline{\text{Conv}}$ désigne l'enveloppe convexe fermée. Si de plus $\mathfrak{X}$ est compact, alors il y a égalité.

Démonstration. • Supposons que $\mathfrak{X}_\lambda \neq \emptyset$. Soit $x_\lambda \in \mathfrak{X}_\lambda$ et soit $\mu \in \mathbb{R}^m$. Alors :

$$\begin{aligned}
\theta(\mu) &= \sup_{x \in \mathfrak{X}} L(x, \mu) \geq L(x_\lambda, \mu) \\
&= f(x_\lambda) + \langle \mu, g(x_\lambda) \rangle \\
&= f(x_\lambda) + \langle \lambda, g(x_\lambda) \rangle + \langle \mu - \lambda, g(x_\lambda) \rangle \\
&= \theta(\lambda) + \langle \mu - \lambda, g(x_\lambda) \rangle.
\end{aligned}$$

Donc, $g(\mathfrak{X}_\lambda) \subset \partial\theta(\lambda)$. D'où le résultat en passant à l'enveloppe convexe fermée.
• Si $\mathfrak{X}$ est compact, d'après le théorème de Weirstrass, $\mathfrak{X}_\lambda$ n'est jamais vide. On admettra qu'il n'existe alors pas de sous-gradient qui ne soit pas dans $\overline{\text{Conv}}\, g(\mathfrak{X}_\lambda)$. □

Théorème 4.2.9 (Dualité forte)

Si λ^* minmise θ et que θ est différentiable en λ^*, alors, pour toute solution x^* du problème primal, $\theta(\lambda^*) = f(x^*)$. Si de plus $\mathfrak{X}_{\lambda^*} \neq \emptyset$, pour tout $x_\lambda \in \mathfrak{X}_{\lambda^*}$, x_λ est solution du problème primal.

Démonstration. Supposons $\mathfrak{X}$ compact pour simplifier la preuve. Alors, $\mathfrak{X}_{\lambda^*} \neq \emptyset$. Soit $x_{\lambda^*} \in \mathfrak{X}_{\lambda^*}$, alors, $g(x_{\lambda^*}) \in \partial\theta(\lambda^*)$. Mais, comme θ est différentiable en λ^*, le sous-différentiel est réduit au singleton $\{\nabla\theta(\lambda^*)\}$. De plus, λ^* minmise θ donc annule le gradient (il n'y a pas de contrainte !). On en déduit finalement que :

$$f(x_{\lambda^*}) \leq \theta(\lambda^*) = L(x_{\lambda^*}, \lambda^*) = f(x_{\lambda^*}) + \langle \lambda^*, g(x_{\lambda^*}) \rangle = f(x_{\lambda^*}) + \langle \lambda^*, \nabla\theta(\lambda^*) \rangle = f(x_{\lambda^*})$$

□

4.3 Dualité pour les problèmes avec contraintes côniques générales

Le problème que l'on considère dans cette section est :

$$(P) \quad : \quad \max_{x \in \mathfrak{X}} f(x), \text{ sous les contraintes : } g(x) \in K$$

où K est un côné. On notera comme d'habitude C l'ensemble des points admissibles. On suppose de plus que toutes les fonctions du problème sont continues et que $\mathfrak{X}$ est une partie de $\mathbb{R}^n$.

Exemple.
1. $K = \{0_{\mathbb{R}^m}\}$: les contraintes sont les mêmes qu'au paragraphe précédent !
2. $K = \mathbb{R}^m_-$: que des contraintes d'inégalités.
3. $K = \{(0, \dots, 0)\} \times \mathbb{R}^{m-k}_-$: contraintes d'inégalités et contraintes d'égalité.
4. $K = \mathbb{S}^+_n$ ensemble des matrices semi-définies positives.

Le problème (P) est en fait clairement équivalent au problème :

$$(\widetilde{P}) \quad : \quad \max_{(x,z) \in \mathfrak{X} \times K} f(x), \text{ sous les contraintes : } g(x) = z \in K$$

On s'est ramené au cas précédent avec uniquement des contraintes d'égalité ! La fonction de Lagrange est alors :

$$\begin{aligned} L_{ext}(x, z, \lambda) &= f(x) + \langle \lambda, g(x) - z \rangle \\ &= f(x) + \langle \lambda, g(x) \rangle - \langle \lambda, z \rangle \end{aligned}$$

Et donc, la fonction duale est :

$$\begin{aligned}\theta_{ext}(\lambda) &= \sup_{(x,z)\in\mathfrak{X}\times K} L_{ext}(x,z,\lambda) \\ &= \sup_{(x,z)\in\mathfrak{X}\times K} f(x) + \langle\lambda, g(x)\rangle - \langle\lambda, z\rangle\end{aligned}$$

Soit $\lambda \in -K^\circ$. Alors, pour tout $z \in K$, $-\langle\lambda, g(x)\rangle \leq 0$. Donc, $\sup(-\langle\lambda, z\rangle) = 0$, où le $\sup$ est obtenu pour $z = 0$ par exemple.

Soit maintenant $\lambda \notin -K^\circ$. Par définition du cône polaire, il existe $z \in K$ tel que le produit scalaire $-\langle\lambda, z\rangle$ est strictement négatif. Considérons alors la suite $(z_n)_{n\in\mathbb{N}}$ définie par $z_n = nz$. Comme K est un cône, cette suite reste dans K. De plus, $-\langle\lambda, z_n\rangle \to +\infty$, ce qui est bien loin du minimum de θ_{ext} ! On a donc montré que :

$$\theta_{ext}(\lambda) = \begin{cases} \sup_{x\in\mathfrak{X}} f(x) + \langle\lambda, g(x)\rangle & \text{si } -u \in K^\circ \\ +\infty & \text{sinon} \end{cases}$$

On peut donc maintenant donner les définitions de la fonction de Lagrange et de la fonction duale dans le cas de contraintes côniques. Ce sont les mêmes qu'au paragraphe précédent ! La seule différence est que le problème duale consiste à minimiser la fonction duale θ non pas sur $\mathbb{R}^m$ mais sur $-K^\circ$.

Définition 4.3.1 (Fonction de Lagrange). La fonction de Lagrange associée au problème (P) est :

$$\begin{array}{rcccl} L & : & \mathfrak{X}\times\mathbb{R}^m & \longrightarrow & \mathbb{R} \\ & & (x,\lambda) & \mapsto & f(x) + \langle\lambda, g(x)\rangle \end{array}$$

Définition 4.3.2 (Fonction duale). La fonction duale du problème (P) est :

$$\begin{array}{rcccl} \theta & : & \mathbb{R}^m & \longrightarrow & \mathbb{R} \\ & & \lambda & \mapsto & \sup_{x\in\mathfrak{X}} L(x,\lambda) \end{array}$$

En résumé :

PRIMAL	DUAL
$\max_{x\in\mathfrak{X}} f(x),\ g(x)\in K$	$\min_{\lambda\in -K^\circ} \theta(\lambda),\ \theta(\lambda) = \sup_{x\in\mathfrak{X}} L(x,\lambda)$

Théorème 4.3.3 (dualité faible)

On a :

$$\inf_{\lambda\in -K^\circ} \theta(\lambda) \geq \sup_{\substack{g(x)=0 \\ x\in\mathfrak{X}}} f(x)$$

Démonstration. La démonstration est la même que celle donnée page 34. □

Remarque. En général, comme on l'a vu dans la section précédente, θ n'est pas différentiable. On a introduit un sous-différentiel $\partial\theta$ pour généraliser la notion de gradient. Dans le problème sans contrainte, on a donné une condition d'optimalité suffisante qui était $0 \in \partial\theta(\lambda^*)$, au lieu du bon vieux $\nabla\theta(\lambda^*) = 0$. Dans le cas avec contraintes, on avait (KKT abstrait) : $\nabla(\lambda^*) \in N_C(\lambda^*)$. Dans le cas non différentiable, cela se généralise à : $0 \in \partial\theta(\lambda^*) + N_C(\lambda^*)$.

Théorème 4.3.4 (dualité forte)

Supposons que λ^* minimise θ sur $-K^\circ$ et que θ est différentiable en λ^*. Alors, pour tout $x_{\lambda^*} \in \mathfrak{X}_{\lambda^*}$, $f(x_{\lambda^*}) = \theta(\lambda^*)$. En particulier, le problème dual est équivalent au problème primal.

Lemme 4.3.5(qui sauve)

Soit K un cône fermé et x un élément de K. Alors, $N_K(x) \subset N_K(0)$.

Démonstration du lemme qui sauve : Soit $g \in N_K(x)$, alors :

$$\forall y \in K,\ \langle g, y - x\rangle \leq 0$$

Soit alors $y \in K$ quelconque. On pose $y_n = y/n$. Comme K est un cône, $y_n \in K$. Donc :

$$\forall n \in \mathbb{N},\ \langle g, n(y_n - x)\rangle \leq 0$$

Donc, en passant à la limite lorsque n tend vers l'infini, on obtient :

$$\langle g, y\rangle \leq 0$$

Autrement dit, $g \in N_K(0)$. □

Démonstration du théorème : On suppose $\mathfrak{X}$ compact pour simplifier la preuve. On a alors : $\overline{Conv}\ g(\mathfrak{X}_{\lambda^*}) = \partial\theta(\lambda)$. Comme θ est différentiable en λ^*, $\{\nabla\theta(\lambda^*)\} = \{g(x_{\lambda^*})\}$. D'où :

$$\nabla\theta(\lambda^*) \in -N_{-K^\circ}(\lambda^*)$$

• Montrons que $\theta(\lambda^*) = f(x_{\lambda^*})$. On sait déjà par le théorème de dualité faible que $\theta(\lambda^*) \geq f(x_{\lambda^*})$. L'autre inégalité s'obtient très simplement en remarquant que :

$$\theta(\lambda^*) = f(x_{\lambda^*}) + \langle\lambda^*, g(x_{\lambda^*})\rangle$$

et que, comme $-g(x_{\lambda^*}) \in N_{-K^\circ}(\lambda^*)$, pour tout $v \in -K^\circ$, et donc en particulier pour $v = 0$, on a :

$$\langle -g(x_{\lambda^*}), v - \lambda^*\rangle \leq 0$$

• Montrons que x_{λ^*} satisfait les contraintes i.e. $g(x_{\lambda^*}) \in K$. On sait que :

$$x_{\lambda^*} \in \mathfrak{X}_{\lambda^*} = \{x \in \mathfrak{X} : \forall y \in \mathfrak{X}, L(x, \lambda^*) \geq L(y, \lambda^*)\}$$

Or, d'après le lemme qui sauve, $N_{-K^\circ}(\lambda^*) \subset N_{-K^\circ}(0) = (-K^\circ)^\circ = -K$. Mais, par optimalité de λ^*, on a :

$$-g(x_{\lambda^*}) \in N_{-K^\circ}(\lambda^*)$$

D'où le résultat. □

4.4 Exemples

4.4.1 Dual d'un programme linéaire

On s'intéresse au problème de programmation linéaire introduit page 21 :

$$\max_{x \in \mathbb{R}^n} c^T x, \quad Ax \leq b$$

La fonction duale est donnée par :

$$L(x, \lambda) = c^T x + \lambda^T (Ax - b), \quad K = \mathbb{R}^m_-$$

Donc :

$$\theta(\lambda) = \sup_{x \in \mathbb{R}^n} \left((c + A^T\lambda)^T x - \lambda^T b\right) = \begin{cases} -\lambda^T b & \text{si } c + A^T \lambda = 0_{\mathbb{R}^n} \\ +\infty & \text{sinon} \end{cases}$$

Le problème dual est celui de minimiser θ sur $-K^\circ = \mathbb{R}^m_-$. On cherche donc :

$$\begin{aligned} \inf_{\lambda \in \mathbb{R}^m_-} \theta(\lambda) &= \inf -b^T \lambda \,, \quad \begin{cases} c + A^T\lambda = 0 \\ \lambda \leq 0 \end{cases} \\ &= \inf b^t \lambda \,, \quad \begin{cases} A^T \lambda = -c \\ \lambda \geq 0 \end{cases} \end{aligned}$$

On remarque que la valeur de x n'intervient pas dans la définition de θ. On peut donc restreindre x à un compact qui contient bien entendu une solution éventuelle du problème de départ. Si, par exemple, $\{x \in \mathbb{R}^n : Ax \leq b\}$ est compact, on peut prendre $\mathfrak{X} = \{x \in \mathbb{R}^n : Ax \leq b\}$.

4.4.2 Relaxation des problèmes combinatoires

On considère les problèmes du type :

$$\min_{x \in \mathbb{R}^n} c^T x, \quad \begin{cases} Ax \leq b \\ x \in \{0; 1\}^n, \text{ ou } x \in \mathbb{Z}^n, \text{ ou } x \in \mathbb{N}^n \end{cases}$$

Une façon de procéder est de traiter $Ax \leq b$ explicitement et les contraintes combinatoires implicitement :

$$\sup_{x \in \mathfrak{X}} c^T x \ , \quad Ax \leq b$$

avec $\mathfrak{X} = \{0;1\}^n$, ou $\mathfrak{X} = \mathbb{Z}^n$, ou $\mathfrak{X} = \mathbb{N}^n$. Ce problème est NP-dur. Considérons le cas où $\mathfrak{X} = \{0;1\}^n$:

$$L(x, \lambda) = c^T x + \lambda^T (Ax - b)$$
$$\theta(\lambda) = \max_{x \in \{0;1\}^n} \left((c + A^T \lambda)^T x - b^T \lambda \right)$$

Si on note $c(\lambda) = c + A^T \lambda$, alors :

$$\begin{aligned} \theta(\lambda) &= \max_{x \in \{0;1\}^n} \left(\sum_{i=1}^n c_i(\lambda) x_i - b^T \lambda \right) \\ &= \sum_{i=1}^n \max(c_i(\lambda), 0) - b^T \lambda \end{aligned}$$

Cette fonction n'est pas différentiable à l'optimum en général. Ce n'est pas étonnant car $\inf \theta$ est un problème simple à résoudre (θ est convexe) mais le problème primal est NP-dur. Si θ était différentiable, on pourrait résoudre facilement un problème NP-dur !

Remarque. On a :

$$\theta(\lambda) = \max_{x \in \{0;1\}^n} \left(\sum_{i=1}^n c_i(\lambda) x_i - b^T \lambda \right) = \max_{x \in [0;1]^n} \left(\sum_{i=1}^n c_i(\lambda) x_i - b^T \lambda \right)$$

Cela signifie que cette relaxation ne fait pas la différence entre $\mathfrak{X} = \{0;1\}^n$ (non convexe) et $\mathfrak{X} = [0;1]^n$ (convexe). Dans le chapitre 5 nous verrons quelques cas où cela n'a pas d'incidence.

Souvent, les problèmes ont une jolie structure de type décomposable :

$$\min_{x \in \mathbb{R}^n} c^T x, \quad \begin{cases} A_1 x \leq b_1 \\ A_2 x \leq b_2 \\ x \in \{0;1\}^n, \text{ ou } x \in \mathbb{Z}^n, \text{ ou } x \in \mathbb{N}^n \end{cases}$$

avec :

$$\min_{x \in \mathbb{R}^n} c^T x, \quad \begin{cases} A_1 x \leq b_1 \\ x \in \{0;1\}^n, \text{ ou } x \in \mathbb{Z}^n, \text{ ou } x \in \mathbb{N}^n \end{cases}$$

facile à résoudre. Dans ces cas, on peut élaborer une relaxation lagrangienne un peu plus subtile :

$$L(x, \lambda) = c^T x + \langle \lambda, A_2 x - b_2 \rangle \ , \quad \mathfrak{X} = \{x \in \mathbb{R}^n : A_1 x \leq b_1\} \cap \{0;1\}^n$$

La fonction duale est alors :

$$\theta(\lambda) = \sup_{x \in \mathfrak{X}} \left((c + A_2^T \lambda)^T x - b_2^T \lambda \right)$$

Cette relaxation est plus fine que de remplacer simplement $\{0;1\}^n$ par $[0;1]^n$ comme on l'a fait précédemment.

4.5 Exercices

Exercice 17. On considère le problème du sac à dos. Etant donnés des biens de valeurs respectives p_i et de volume v_i pour $i = 1, \ldots, n$, on veut rassembler les biens constituant un butin de valeur maximale sous une contrainte de volume donné V. On se donne les valeurs suivantes :

$$p = \begin{bmatrix} 10 \\ 4 \\ 14 \end{bmatrix} \text{ et } v = \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}$$

et on prendra $V = 4$.

1. Donner la relaxation lagrangienne du problème.
2. Tracer la fonction duale $\theta(\lambda)$.
3. Donner la solution optimale duale puis proposer une solution primale.

Exercice 18. On considère le problème d'affectation avec contrainte de budget. Etant donné un ensemble de machines M et un ensemble de taches N, de même cardinal, il faut affecter une machine à chaque tache. Le temps employé par la machine i à la tache j est de t_{ij}. Le coût de cette affectation est de c_{ij}. On ne veut pas dépasser un coût total de C. Le but est de choisir une affectation qui dure le moins longtemps avec un coût inférieur ou égal à C.

1. une formulation de ce problème sous la forme d'un problème linéaire avec contraintes binaires.
2. Proposer une relaxation lagrangienne de ce problème.

Exercice 19. On considère un ensemble de dépôts $N = \{1, \ldots, n\}$ et un ensemble de clients $M = \{1, \ldots, m\}$. On suppose qu'il y a un coût fixe f_j correspondant au fait d'utiliser le dépôt j. Il y a aussi un coût de transport c_{ij} pour amener le produit du dépôt j au client i. Le problème est de déterminer quels dépôts ouvrir et parmi ceux que l'on va ouvrir, quel dépôt fournit quel client. Ces choix sont faits de manière à minimiser le coût total.

Ce problème est souvent appelé uncapacitated facility location (UFL) dans la littérature anglaise sur le sujet.

On modélise le problème en prenant pour variables les $x_{ij} \in \{0, 1\}$ indiquant que le client i est servi par le dépôt j. On modélisera l'ouverture du dépôt j par une variable $y_j \in \{0, 1\}$. On suppose de plus que la demande correspondant au client i est de 1.

1. Quelle relation existe-t-il entre les x_{ij} et les y_j ?
2. Donner une formulation du problème.
3. Proposer une relaxation lagrangienne.
4. On considère l'exemple avec $n = 5$, $m = 6$, et avec la matrice des coûts de transport :

$$C = \begin{bmatrix} 6 & 4 & 3 & 2 & 1 & 3 \\ 2 & 10 & 2 & 0 & 8 & 2 \\ 1 & 2 & 4 & 4 & 6 & 4 \\ 3 & 6 & 1 & 1 & 2 & 8 \\ 5 & 1 & 3 & 4 & 5 & 1 \end{bmatrix} \text{ et } f = \begin{bmatrix} 4 \\ 8 \\ 11 \\ 7 \\ 5 \end{bmatrix}$$

5. Calculer la valeur de $\theta(\lambda)$ pour $\lambda = [5, 6, 3, 2, 6, 4]^t$.
6. Pour quel x obtient-on cette valeur ?
7. Comment modifier x pour obtenir une solution primale à peu près satisfaisante ?

4.6 Correction des exercices

La fonction à maximiser est :

$$f(x) = -\sum_{i,j} c_{i,j}x_{i,j} - \sum_j y_j f_j$$

sous les contraintes :

$$x \in \{0;1\}^{mn}, y \in \{0,1\}^m$$
$$\forall i,j : x_{i,j} \leq y_j$$
$$\forall i : \sum_j x_{i,j} = 1$$

La fonction de Lagrange est donc :

$$\begin{aligned} L(x,\lambda) &= -\sum_{i,j} c_{i,j}x_{i,j} - \sum_j y_j f_j + \sum_i \lambda_i \left(\sum_j x_{i,j} - 1\right) \\ &= \sum_j \left(\sum_i (\lambda_i - c_{i,j})x_{i,j} - y_j f_j\right) - \sum_i \lambda_i \end{aligned}$$

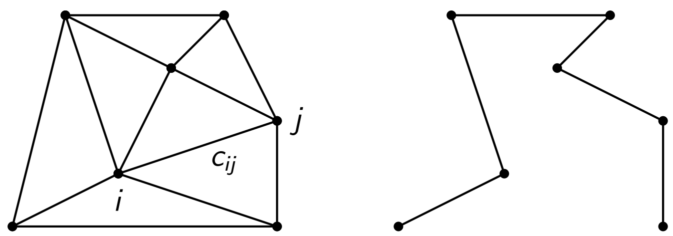
Chapitre 5

Problèmes combinatoires

5.1 Introduction

Commençons par présenter rapidement quelques exemples de problèmes combinatoires. Cette liste n'est bien sûr pas exhaustive, on pourra par exemple citer le problème de coloration mais aussi les problèmes du max cut, vertex cover, independence set...

5.1.1 Arbre couvrant (minimum spanning tree)

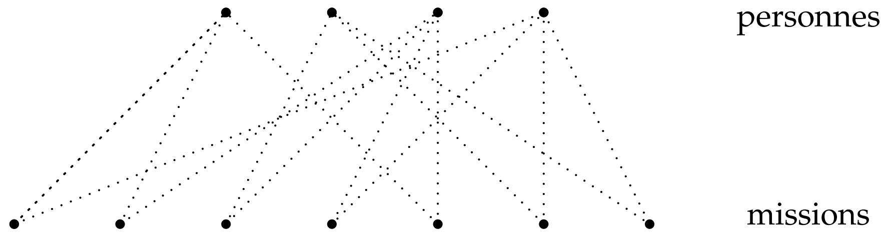


On considère un graphe à poids (verts ou rouges). On veut relier les noeuds de manière la plus économique possible sous la contrainte que tous les noeuds du graphe soient reliés (indirectement) entre eux. On peut penser par exemple à un réseau d'égoûts. On cherche donc à construire ce qu'on appelle un "arbre couvrant" de poids minimal, c'est-à-dire un sous-graphe tel qu'il n'existe pas de boucle et dont tous les noeuds sont reliés. C'est le problème du "minimum spanning tree".

5.1.2 Plus court chemin (shortest path problem)

On considère un graphe à poids et deux noeuds i et j de ce graphe. Le problème du plus court chemin est celui de trouver un chemin allant de i à j de poids minimal. C'est un problème très fréquent qu'on retrouve par exemple dans les réseaux informatique. Dans la littérature on le retrouve sous le doux nom de "shortest path problem".

5.1.3 Problème d'affection, problème du mariage (assignment problem)



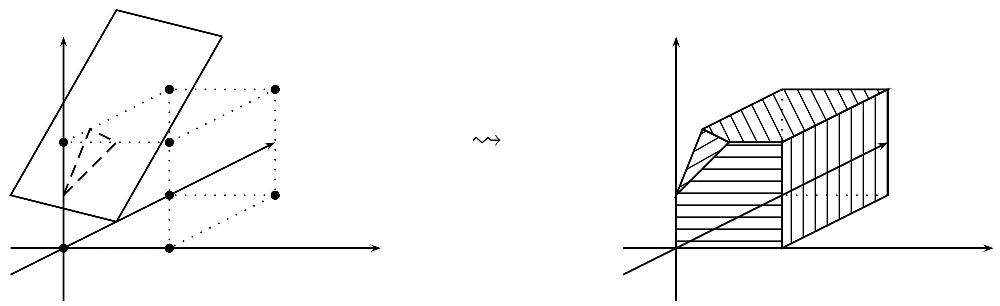
On considère un graphe biparti : les noeuds sont séparables en deux sous-ensembles et les arêtes ne peuvent joindre que des noeuds de sous-ensembles différents. Ce genre de situations arrive par exemple quand on souhaite affecter des équipages d'avion à des trajets, ou encore des tâches à des machines. Chaque arête a un poids et il faut trouver un ensemble d'arêtes non adjacentes de poids minimal tel que toutes les personnes soient employées ou toutes les tâches soient effectuées suivant qu'il y a plus de personnes que de missions ou le contraire. On appelle ce problème "assignment problem", ou "matching problem" dans le cas où il y a autant de personnes que de missions.

5.2 Exactitude de la relaxation pour certains problèmes combinatoires

On considère le problème :

$$\max_{x \in \mathbb{R}^n} c^T x, \quad \begin{cases} Ax - b \in K \\ x \in \{0;1\}^n \text{ ou } x \in \mathbb{Z}^n \end{cases}$$

où $K = \{0\}^{n_e} \times \mathbb{R}_-^{n_i}$. La question est de savoir quand est-ce que la relaxation présentée au chapitre 4 donne la bonne solution. On cherche donc dans quelles situations la relaxation de $\{0;1\}^n$ en $[0;1]^n$ (ou de $\mathbb{Z}^n$ en $\mathbb{R}^n$) ne change pas le problème.



Définition 5.2.1 (totale unimodularité). Une matrice est dite totalement unimodulaire si et seulement si tous ses mineurs (i.e. les déterminants de ses sous-matrices carrées) valent tous $-1, 0$ ou 1.

Rappel (Formule de Cramer). Si $A \in M_n(\mathbb{R})$ est une matrice inversible et si $b \in \mathbb{R}^n$, alors, la solution du système $Ax = b$ est donnée par :

$$\forall i \in [\![1, n]\!] \ : \ x_i = \frac{\det A_{-i,b}}{\det A}$$

où $A_{-i,b}$ est la matrice dont les colonnes sont celles de A sauf la i-ème qui a été remplacée par b.

Théorème 5.2.2

Soit A une matrice totalement unimodulaire. Alors, la solution (si elle existe et si elle est finie) du problème :

$$\max_{x \in \mathbb{R}^n} c^T x, \ \ Ax - b \in K, \ \ b \in \mathbb{Z}^n$$

est entière.

Démonstration. Soit x^* optimal. Il vérifie donc les équations de KKT. Rappelons que l'on a noté $\mathcal{J}_{x^*}$ l'ensemble des contraintes actives en x^*. On a alors :

$$-c \in N_{\{Ax-b \in K\}}(x^*) = \{\sum_{j \in \mathcal{J}_{x^*}} \lambda_j a_j, \ \forall j \in [\![n_e+1, n_e+n_i]\!] \ : \ \lambda_j \geq 0\}$$

On notera dans la suite $(a_1, \cdots, a_n)$ les lignes de la matrice A.
• Supposons dans un premier temps que $|\mathcal{J}_{x^*}| = n$ et que la famille $(a_1, \cdots, a_n)$ est libre. Alors :

$$\forall j \in \mathcal{J}_{x^*} \ : \ a_j^T x - b_j = 0$$

Donc, en utilisant la formule de Cramer, x est à coordonnées entières.
• Si la famille $(a_1, \cdots, a_n)$ est liée, on élimine un vecteur (notons-le a_{i_0}) qui est combinaison linéaire des autres. Soit alors $v \in KerA_{-i_0}$ (A_{-i_0} est la matrice A à laquelle on a retiré la i_0-ième ligne). Deux cas se présentent :

- $\{x + tv, t \in \mathbb{R}^n\} \subset \{Ax - b \in K\}$. Dans ce cas, $\{Ax - b \in K\}$ n'est pas borné ce qui est absurde.
- $\exists j \in \mathcal{J}_{x^*}, \ \exists t \in \mathbb{R} \ : \ a_j^T(x + tv) - b_j = 0$. On intègre alors cet indice à la famille des a_i et on recommence jusqu'à ce que $|\mathcal{J}_{x^*}| = n$ et que $(a_1, \cdots, a_n)$ soit libre.

□

Définition 5.2.3 (matrice d'incidence d'un graphe). Soit G un graphe dont les noeuds sont numérotés de 1 à n et les arêtes de 1 à m. La matrice d'incidence de G est la matrice $n \times m$ indiciée par les noeuds sur les lignes et les arêtes sur les colonnes et dont l'indice i, j vaut 1 si l'arête j est reliée au noeud i. Dans le cas de graphes orientés, la matrice d'incidence est construite de la même manière sauf que l'indice i, j vaut -1 si l'arête j part du noeud i et 1 si l'arête j arrive au noeud i.

Théorème 5.2.4

Soit G un graphe et A sa matrice d'incidence. Alors, A est totalement unimodulaire si et seulement si G est biparti.

Théorème 5.2.5

Soit G un graphe orienté et A sa matrice d'incidence. Alors, A est totalement unimodulaire.

Démonstration. On procède par récurrence sur la taille des sous-matrices carrées de A :
• Les mineurs de taille 1 d'une matrice d'incidence valent évidemment -1, 0 ou 1 puisqu'une matrice d'incidence n'est composée que de -1, de 0 et de 1.
• Supposons le résultat vrai pour les mineurs de taille $k \times k$ et soit M une sous-matrice de A de taille $(k+1) \times (k+1)$. Trois cas se présentent :

- Toutes les colonnes de M contiennent un 1 et un -1. Alors :

  $$(1, \cdots, 1)M = 0$$

  Donc, M n'est pas inversible. Ainsi, $\det M = 0$.
- Il existe une colonne de M qui est nulle. Alors, $\det M = 0$.
- Il existe dans M une colonne où il y a un 1 ou un -1 et que des zéros. En développant par rapport à cette colonne, le résultat est encore vrai par hypothèse de récurrence.

□

5.3 Exemples

5.3.1 Problème d'affectation

On suppose ici qu'il y a plus de missions que de personnes pour fixer les idées. On rappelle que le problème est alors :

$$\min \sum_{i,j} c_{i,j} x_{i,j} \ , \quad \begin{cases} \displaystyle\sum_j x_{i,j} \geq 1 & \text{(chacun a un job)} \\ \displaystyle\sum_i x_{i,j} = 1 & \text{(chaque job n'est fait qu'une seule fois)} \end{cases}$$

où on a noté :

$$x_{i,j} = \begin{cases} 1 & \text{si on choisit l'arête } (i,j) \\ 0 & \text{sinon} \end{cases}$$

Si on note A la matrice d'incidence du graphe où les n_e premières lignes correspondent aux personnes et les n_i suivantes aux jobs, alors on peut réécrire le problème sous la forme :

$$\min c^T x \ , \quad Ax - (1, \cdots, 1)^T \in K = \{0; 1\}^{n_e} \times \mathbb{R}_-^{n_i} \ , \quad x \geq 0$$

Comme A est totalement unimodulaire, les solutions éventuelles sont entières. De plus, $x \in \{0; 1\}^n$ car la somme $\sum c_{i,j} x_{i,j}$ est minimale et les $c_{i,j}$ sont positifs.

5.3.2 Plus court chemin

On reprend le problème du plus court chemin introduit page 43. On note A la matrice d'incidence du graphe orienté où le premier noeud est i et le dernier est j. Le problème s'écrit alors :

$$\min \sum_{i,j} c_{i,j} x_{i,j} \ , \quad Ax = (-1, 0, \cdots, 0, 1)^T, \quad x \geq 0$$

Comme la matrice A est totalement unimodulaire, comme précédemment, les solutions éventuelles sont entières et $x \in \{0; 1\}^n$ car les $c_{i,j}$ sont positifs.

Chapitre 6

Scilab

6.1 Introduction

6.1.1 Présentation

Scilab est un logiciel de calcul numérique. Il a été développé par des chercheurs de l'INRIA (Institut National De Recherche en Informatique et en Automatique) et de l'ENPC (Ecole Nationale des Ponts et Chaussées) depuis 1990. Depuis 2003 il est développé et maintenu par l'INRIA. Scilab est un logiciel "semi-libre" : son code source est disponible gratuitement mais sa licence n'autorise pas la distribution commerciale d'une version modifiée. Scilab est bien sûr multiplateforme. Il fonctionne aussi bien sous Linux, Mac OS ou Windows. Il est disponible au téléchargement sur **www.scilab.org** et sur les dépôts officiels de nombreuses distributions linux.

6.1.2 Les bases

Pour faire de l'optimisation avec Scilab, il suffit de connaître quelques commandes de base. Pour vous familiariser avec la bête, essayez donc d'entrer ces quelques commandes :

```
--> 2+2
--> a=2
--> a=a+1
--> a=a+1;
--> a
```

Un des atouts de Scilab est la simplicité de l'écriture matricielle. Une matrice s'écrit avec des crochets. Un changement de colonne est déclaré avec un espace ou une virgule, et un changement de ligne avec un point virgule. L'écriture de matrice par bloc est alors très aisée :

```
--> A=[1 2 ; 3 4]
--> B=[-1 -4 ; 6 -2]
--> C=[A B]
--> D=[A ; B]
--> E=D'
```

Pour accéder à un élément dans une matrice, on utilise des parenthèses. L'extraction d'un bloc est aussi une opération des plus simples :

```
--> A(1,2)
--> A(1,$)
--> A(1,2)=8
--> A(:,1)
--> C(:,3:4)
```

Comme tous les langages de programmation, Scilab offre la possibilité d'effectuer des boucles, du conditionnement... Voici un exemple simple :

```
--> res=[]
--> for k=-1:0.2:2
-->   if k>0 then
-->      res=[res ; k+1]
-->   else
-->      res=[res ; k/2]
-->   end
--> end
```

Notez au passage que l'indentation n'est pas obligatoire. Cependant elle est conseillée car elle rend vos programmes beaucoup plus lisibles. Si vous voulez enregistrer votre travail, vous pouvez utiliser l'éditeur Scipad en cliquant sur le bouton "Editor". Maintenant, écrivons une fonction. Pour cela, démarrez Scipad et écrivez :

```
//Ceci est un commentaire
function [x,y]=mafonction(a,b)
   x=a+b
   y=a*b
endfunction
```

Enregistrez votre travail et cliquez sur "Execute -> Load into Scilab". Vous pouvez maintenant utiliser votre fonction depuis Scilab en l'appelant par son petit nom :

```
--> [somme,produit]=mafonction(2,3)
```

Avant d'en finir avec ce paragraphe d'introduction, donnons une liste (non exhaustive) de quelques fonctions qui sont très utiles :

help	affiche l'aide
help topic	affiche l'aide relative à topic
size(A)	donne la taille de la matrice A.
plot2d(x,y)	trace une ligne brisée.
ones(n,m)	matrice $n \times m$ remplie de 1.
zeros(n,m)	matrice $n \times m$ remplie de 0.
eye(n,m)	matrice $n \times m$ avec des 1 sur la diagonale (0 ailleurs).

6.2 Scilab et l'optimisation

6.2.1 La fonction linpro

Scilab dispose de la fonction linpro pour résoudre les problèmes de programmation linéaire. Elle peut être appelée de plusieurs manières selon ce qu'on cherche à résoudre. Par exemple :

```
[s,lagr,fs]=linpro(c,C,rhs)
```

résout le problème de programmation linéaire :

$$\min_x c^T x, \quad \text{sous la contrainte : } Cx \leq rhs$$

Notez qu'il est inutile de spécifier à Scilab la dimension du problème, les données la rendant implicite. Le résultat renvoyé est un vecteur. La première coordonnée de ce vecteur est une solution du problème. La deuxième est le vecteur des multiplicateurs de Lagrange correspondant. Enfin, la troisième est la valeur de la fonction à minimiser en ce point.

Bien que l'on puisse écrire tous les problèmes de programmation linéaire sous cette forme, pour simplifier l'utilisation de la fonction linpro il est possible d'ajouter des paramètres. Ainsi, si on veut entrer p contraintes d'égalité, une borne inférieure (lb) et une borne supérieure (ub), le problème s'écrit :

$$\min_x c^T x, \quad \text{sous les contraintes : } \begin{cases} C(j,:)x = rhs & \text{si } j \leq p \\ C(j,:)x \leq rhs & \text{sinon} \\ lb \leq x \leq ub \end{cases}$$

et se résout avec Scilab de cette façon :

```
[s,lagr,fs]=linpro(c,C,b,lb,ub,p)
```

où on aura bien pris soin d'écrire les contraintes d'égalité dans les p premières lignes de la matrice C. Il peut arriver qu'il y ait une borne inférieure mais pas de borne supérieure (ou le contraire). On peut dans ce cas donner pour ub un tableau vide ([]). Si il y a une borne supérieure qui ne porte pas sur toutes les coordonnées de x on peut donner comme borne supérieure $+\infty$ pour ces composantes (en utilisant au choix %inf ou number_properties('huge')). Pour aller plus loin, consultez l'aide de la fonction linpro en tapant sous Scilab :

```
--> help linpro
```

6.2.2 La fonction quapro

La fonction quapro s'utilise de la même manière que la fonction linpro. Elle prend en argument un élément de plus : la matrice de la forme quadratique. Ainsi, pour résoudre le problème de programmation quadratique suivant :

$$\min_x \left(\frac{1}{2} x^T Q x + q^T x \right) \quad \text{sous la contrainte : } Cx \leq rhs$$

on écrit :

```
[s,lagr,fs]=quapro(Q,q,C,rhs)
```

On peut ajouter des arguments supplémentaires exactement de la même manière qu'avec la fonction linpro. Pour aller plus loin, consultez l'aide de Scilab :

```
--> help quapro
```

6.3 Exercices

Utilisation de la fonction linpro

Résoudre les exercices suivants en utilisant la fonction LINPRO de Scilab

Exercice 20. On reprend l'exercice 6 introduit page 18. On cherche à minimiser la fonction de $\mathbb{R}^2$ suivante :

$$f(x,y) = -2x + y$$

On suppose de plus que x et y sont soumis aux contraintes :

$$\begin{aligned} 2x + 3y &\geq 6 \\ -x + y &\leq 3 \\ x &\leq 2 \\ x \geq 0 \quad y &\geq 0 \end{aligned}$$

Exercice 21. Une entreprise doit transporter du beurre de cacahuète entre trois dépôts D_1, D_2 et D_3 et trois magasins M_1, M_2 et M_3. Chaque dépôt a une quantité limitée de marchandise, chaque destination une demande, et le coût unitaire pour transporter un camion de beurre de cacahuète varie suivant le dépôt et le magasin. On résume tout dans le tableau ci-dessous (la dernière colonne représente le coût unitaire) :

Dépôt	Quantité disponible	Magasin	Demande	⁄	M_1	M_2	M_3
D_1	35	M_1	45	D_1	5	10	10
D_2	40	M_2	50	D_2	20	30	20
D_3	40	M_3	15	D_3	5	8	12

Exercice 22. Un bûcheron a 100 hectares de bois de feuillus. Couper un hectare de bois et laisser la zone se régénérer naturellement coûte 10 k$ par hectares, et rapporte 50 k$. Alternativement, couper un hectare de bois, et replanter avec des pins coûte 50 k$ par hectares, et rapporte à terme 120 k$. Sachant que le bûcheron n'a que 4000 k$ en caisse au début de l'opération, déterminer la meilleure stratégie à adopter et le profit escomptable.

Exercice 23. Trois machines M_1, M_2 et M_3 peuvent produire chacune deux types de pièces P_1 et P_2. Le temps de fabrication d'un pièce P_i sur la machine M_j est donné comme suit :

(temps de fab. de P_1) sur M_1 : 3h , sur M_2 : 4h sur M_3 : 4h
(temps de fab. de P_2) sur M_1 : 4h , sur M_2 : 6h sur M_3 : 5h

On veut fabriquer au moindre coût 6 pièces de type P_1 et 8 pièces de type P_2. La machine M_1 est disponible 14 heures et les machines M_2 et M_3 sont disponibles chacune 24 heures. Le cout horaire de M_1 est 7, celui de M_2 est 5 et celui de M_3 est 6.

Régression linéaire, régression ridge, LASSO

Exercice 24 (Rappels sur la régression linéaire). On se propose d'étudier un problème de régression a priori intraitable. La variable observée est Y et on souhaite expliquer cette variable par un ensemble de co-variables $X_1, \ldots, X_p$. Le modèle linéaire est donc

$$Y = a_0 + a_1 X_1 + \cdots + a_p X_p + \varepsilon \tag{6.1}$$

où ε est une variable gaussienne à valeurs dans $\mathbb{R}$ $\mathcal{N}(0, \sigma^2)$. Il s'agit donc d'estimer les coefficients réels $a_0, a_1, \ldots, a_p$. Pour cela, on effectue plusieurs mesures et on obtient autant de variables que d'expériences réalisées, c'est à dire

$$Y_i = a_0 + a_1 X_{i,1} + \cdots + a_p X_{i,p} + \varepsilon_i \qquad i = 1, \ldots, n. \tag{6.2}$$

Afin d'utiliser l'écriture matricielle, on note $Y = [Y_1, \ldots, Y_n]^t$, $\varepsilon = [\varepsilon_1, \ldots, \varepsilon_n]^t$ et $X = [(X_{ij})_{i=1,\ldots,n\, j=0,\ldots,p}]$ avec $X_{i,0} = 1$ quelque soit i et on oublie l'ancien Y et l'ancien ε dont on ne se resservira plus par la suite car ce qui nous interesse est évidemment d'estimer les $a_0, a_1, \ldots, a_p$. Les équations (6.2) deviennent alors avec ces notations

$$Y = X\, a + \varepsilon. \tag{6.3}$$

1. Rappeler la formule donnant l'estimateur au sens des moindres carrés
2. Cet estimateur au sens des moindres carrés est solution du problème d'optimisation

$$\min_{a \in \mathbb{R}^{p+1}} \|Y - Xa\|_2^2 \tag{6.4}$$

où $\|\cdot\|_2$ dénote la norme euclidienne. Calculer la matrice hessienne de cette fonction à minimiser. Ce problème d'optimisation est-il convexe ? Pouvez vous retrouver la formule de l'estimateur au sens des moindres carrés en résolvant ce problème d'optimisation ?

Le problème que l'on souhaite aborder après ces rappels est celui de trouver les coefficients $a_0, \ldots, a_p$ dans le cas où $p \geq n$ voire bien plus grand que n. Ce cas arrive en fait très souvent en pratique et est donc impossible à traiter par la minimisation de (6.4) car la matrice hessienne est dans ce cas dégénérée, c'est à dire que son noyau est non réduit à $\{0\}$.

3. Pourquoi cette dernière remarque est-elle vraie ? Combien y'a-t-il de solution dans ce cas au problème (6.4) ?

Exercice 25 (Régression Ridge). Pour parer au problème soulevé dans l'exercice précédent, deux approches sont possibles. La première s'appelle la ridge regression et consiste à chercher a comme le vecteur qui minimise

$$\min_{a \in \mathbb{R}^{p+1}} \|Y - Xa\|_2^2 + \lambda \|a\|_2^2. \tag{6.5}$$

1. Ce problème est-il convexe ? admet-il une solution unique ?
2. Implanter cette solution à l'aide de la fonction `quapro` pour les données "detroit" qui donnent le nombre d'homicides en fonction de divers paraèmetres dans la ville de Detroit de 1961 à 1970. Faites varier le paramètre λ et affichez sur la même figure l'évolution jointe des composantes de a en fonction de λ.

Exercice 26 (Méthode du LASSO). Une autre approche pour le problème précédent consiste à utiliser le LASSO, une technique dont le succès est due à l'étude mathématique et à la promotion ultérieure qu'en a fait Robert Tibshirani de l'université de Stanford en Californie. Cette méthode consiste à remplacer la pénalisation quadratique par une pénalisation utilisant la norme l_1 du vecteur a, et revient donc à résoudre le problème

$$\min_{a \in \mathbb{R}^{p+1}} \|Y - Xa\|_2^2 + \lambda \|a\|_1. \tag{6.6}$$

1. Montrer de manière intuitive que ce problème peut se mettre sous la forme

$$\min_{a,\ u \in \mathbb{R}^{p+1}} \|Y - Xa\|_2^2 + \lambda \sum_{k=1}^{p} u_k \text{ sous les contraintes } -u_k \leq a_k \leq u_k \quad k = 1, \ldots, p. \tag{6.7}$$

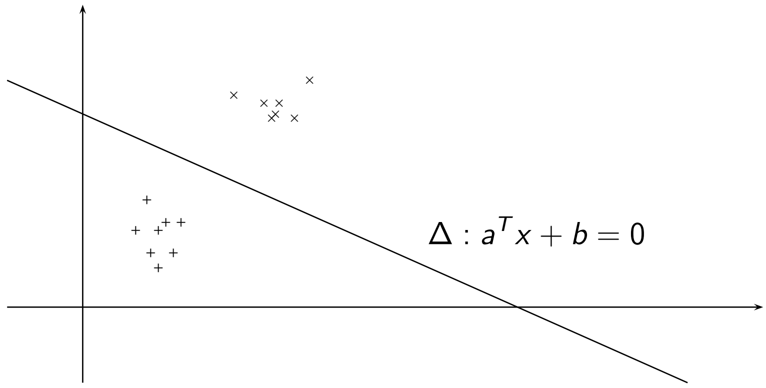
2. Implanter cette solution à l'aide de la fonction quapro pour les données "detroit" qui donnent le nombre d'homicides en fonction de divers paraèmetres dans la ville de Detroit de 1961 à 1970. Faites varier le paramètre λ et affichez sur la même figure l'évolution jointe des composantes de a en fonction de λ.

Remarque. à partir d'une valeur assez grande de λ, certaines composantes de a deviennent nulles. Ainsi, la méthode effectue un choix automatique des variables importante pour expliquer le phénomène observé. On résout alors ce qu'on appelle la question du choix de modèle d'une manière globale. L'étude théorique de cette méthode est actuellement en pleine effervescence (voir par exemple http ://www.acm.caltech.edu/ emmanuel/papers/LassoPredict.pdf parmi plein d'autres). Cette technique est aussi extrêment utlisée pour la détection des groupes de gènes important pour une maladie spécifique étudiée mais dans ce cas, le modèle introduit n'est plus une régression linéaire mais une régression logistique, ce qui donne un problème d'optimisation un peu plus compliqué ! Un article intéressant à ce sujet est http ://www-stat.stanford.edu/ hastie/Papers/JRSSB.69.4%20(2007)%20677%20Park.pdf parmi une tonne d'autres bien sûr.

Support Vector Machines linéaires pour la classification

Le problème des Support Vector Machines a été introduit dans la section 3.2.3 à la page 25. Le problème était alors de trouver le meilleur hyperplan de séparation de deux groupes de points. On a alors introduit des variables y_i et posé $y_i = 1$ si x_i est dans le premier groupe de points et $y_i = -1$ sinon. Dans le cas où les données sont séparables, on a vu qu'un hyperplan d'équation $a^T x + b = 0$ était optimal si et seulement si (a, b) résolvait le problème quadratique suivant :

$$\min_{(a,b)} \|a\|^2, \quad \forall i : \ y_i(a^T x_i + b) \geq 1$$



On va générer des données artificiellement. On tire pour cela 10 vecteurs gaussien bidimensionnels d'espérance le vecteur $(0; 0)$ et de variance l'identité et 10 vecteurs gaussien bidimensionnels également d'espérance $(10, 5)$ et de variance 2 fois l'identité. Sous Scilab, la fonction pour tirer ces points au hasard est grand avec l'option mn comme "multivariate normal".

1. Représentez les points tirés dans le plan à l'aide de la fonction plot2d. Les deux groupes de points sont-ils bien séparables ? Si non, recommencez l'opération jusqu'à ce que ça le soit ! Cela ne sera pas trop long. Enfin, j'espère ...
2. Résolvez le problème de séparation pour ces données artificielles.
3. Tracez la droite de séparation optimale que vous avez trouvée.
4. Simulez une centaine de nouveaux vecteurs gaussiens avec les mêmes paramètres. Combien sont mal classés ?

Support Vector Machines avec débordement

Le gros problème avec la recherche d'un hyperplan séparateur est que souvent, les données ne sont tout simplement pas séparables ! il n'y a alors pas de solution au problème du learning (vu page 25). Une façon raisonnable de généraliser le problème est la suivante : au lieu d'imposer une marge supérieure ou égale à 1, on peut autoriser qu'elle soit supérieure à $(1 - \xi_i)$ pour chaque observation $i \in [\![1, m]\!]$, avec ξ_i le plus petit possible tout en restant supérieur ou égal à zéro. Si les données sont séparables, on pourra donc prendre tous les $\xi_i = 0$ nuls. Cela peut se mettre sous la forme :

$$\min_{(a,b,\xi)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}^m} \|a\|_2^2 + \lambda\|\xi\|_2^2, \quad \text{sous les contraintes} \left\{ \begin{array}{l} \forall i \in [\![1, m]\!] \ : \ y_i(a^t x_i + b) \geq 1 - \xi_i \\ \xi \geq 0 \end{array} \right.$$

Dans cet exercice nous allons utiliser des données bla bla bla...

1. Expliquer pourquoi effectivement ce problème est équivalent au problème du learning dans le cas de données séparables.
2. Implémenter cette méthode pour les données de cancer du sein où les colonnes correspondent aux variables 1. Sample code number (id number), 2. Clump Thickness (1 - 10), 3. Uniformity of Cell Size (1 - 10) 4. Uniformity of Cell Shape (1 - 10) 5. Marginal Adhesion (1 - 10), 6. Single Epithelial Cell Size (1 - 10), 7. Bare Nuclei (1 - 10),

8. Bland Chromatin (1 - 10), 9. Normal Nucleoli (1 - 10), 10. Mitoses (1 - 10), 11. Class (2 for benign, 4 for malignant).

   Les deux classes à identifier sont les tumeurs benignes et les malignes (dernière variable).

3. Enlever quelques données (une dizaine) de la base puis construire l'hyperplan sur les données restantes. Combien sont mal classées ?
4. Une méthode systématique d'étude consiste en le fait d'enlever une donnée, de calculer l'hyperplan et de voir si cette donnée est oui ou non bien classée et de recommencer en tirant à chaque fois une donnée différente. Cela s'appelle la validation croisée. Implémenter cette méthode. Quel pourcentage de mauvais classement obtenez vous ?

6.4 Correction des exercices

Support Vector Machines linéaires pour la classification

On commence par générer des réalisations de vecteurs gaussiens avec les données de l'énoncé. Attention ! On ne veut pas avoir de nouvelles réalisations à chaque exécution du programme ! Veillez donc à faire cette manipulation sous Scilab directement et pas sous Scipad.

```
--> groupe1 = grand(10,'mn',[0;0],eye(2,2));
--> groupe2 = grand(10,'mn',[10;5],2*eye(2,2));
```

Pour vérifier que les données obtenues sont bien séparables, on utilise la fonction plot2d. On ajoute un troisième argument pour changer le "style" du graphique. En effet, on ne veut pas dessiner des lignes brisées mais bien un nuage de point. Il faut pour cela donner un entier négatif.

```
--> plot2d(groupe1(1,:) , groupe1(2,:) , -1)
--> plot2d(groupe2(1,:) , groupe2(2,:) , -2)
```

La droite de séparation recherchée a pour équation $\Delta : a^T x + b = 0$. Si on note z le vecteur $(a, b) \in \mathbb{R}^3$, on a vu (cf page 27) que le problème à résoudre pouvait s'écrire :

$$\min_{z\in\mathbb{R}^3} \frac{1}{2} z^T Qz + q^T z \ , \quad Cx \leq rhs$$

où :

$$Q = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \ , \ q = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \ , \ C = \begin{pmatrix} -x_i^T - 1 \\ \vdots \\ x_i^T + 1 \end{pmatrix} \ , \ rhs = \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix}$$

Il ne nous reste alors plus qu'à écrire ces matrices en Scilab et la fonction quapro fera le reste.

```
--> Q = [2 0 0 ; 0 2 0 ; 0 0 0]
--> q = zeros(3,1)
--> C = [-groupe1' -ones(10,1) ; groupe2' ones(10,1)]
--> rhs = -ones(20,1)
--> [z,lagr,fz] = quapro(Q,q,C,rhs)
--> a = z(1:2)
--> b = z(3)
```

Et voilà ! Un simple plot2d permet de se convaincre que le résultat donné par Scilab est correct. Amusons-nous maintenant à regarder ce qu'il se passe si on effectue d'autres tirages aléatoires : sont-ils bien classés ?

```
--> u=grand(50,'mn',[0;0],eye(2,2));
--> v=grand(50,'mn',[10;5],2*eye(2,2));
--> length(find(a'*u+b>0))+length(find(a'*u+b<0))
```

Support Vector Machines avec débordement

La principale difficulté de cet exercice est, comme souvent avec Scilab, d'écrire le problème sous forme “standard”, i.e. sous la forme :

$$\min_{x\in\mathbb{R}^k}\left(\frac{1}{2}x^TQx+q^Tx\right),\quad \text{sous les contraintes } \begin{cases} Cx\le rhs\\ lb\le x\le ub\end{cases}$$

Il s'agit donc de trouver la matrice Q, le vecteur q, la matrice de contraintes C, le vecteur rhs ainsi que les vecteurs lb et ub. Une fois les données séparées en deux sous-groupes, si on note n_1 le nombre d'éléments dans le premier groupe et n_2 le nombre d'éléments dans le second, on peut écrire :

$$Q(\lambda)=\begin{pmatrix} 2 &&&&&\\ &\ddots&&&&\\ &&2&&&\\ &&&0&&\\ &&&&2\lambda&&\\ &&&&&\ddots&\\ &&&&&&2\lambda\end{pmatrix},\quad q=\begin{pmatrix}0\\ \vdots\\ \vdots\\ \vdots\\ \vdots\\ 0\end{pmatrix},\quad lb=\begin{pmatrix}-\infty*1_n\\ -\infty\\ 0\\ \vdots\\ \vdots\\ 0\end{pmatrix}$$

$$C=\begin{pmatrix} x_1^T & 1 & -1 & & & 0 & \dots & 0\\ \vdots & \vdots & & \ddots & & \vdots & & \vdots\\ x_{n_1}^T & 1 & & & -1 & 0 & \dots & 0\\ -x_{n_1+1}^T & -1 & 0 & \dots & 0 & -1 & &\\ \vdots & \vdots & \vdots & & \vdots & & \ddots &\\ -x_{n_1+n_2}^T & -1 & 0 & \dots & 0 & & & -1\end{pmatrix},\quad rhs=\begin{pmatrix}-1\\ \vdots\\ \vdots\\ \vdots\\ \vdots\\ -1\end{pmatrix}$$

où 1_n est le vecteur $(1,\dots,1)^T\in\mathbb{R}^n$ et où on a noté $(x_1^T,\dots,x_{n_1}^T)$ les lignes de groupe1 et $(x_{n_1+1}^T,\dots,x_{n_1+n_2}^T)$ celles de groupe2. La matrice C est une matrice $(n_1+n_2)\times(n+1+n_1+n_2)$.

En Scilab, ∞ s'écrit %inf. Comme il n'y a pas de borne supérieure sur les variables, on peut rentrer un tableau vide pour *ub*. Heureusement, il est beaucoup plus aisé d'écrire ces matrices sous Scilab (qui accepte l'écriture matricielle par bloc) qu'il ne l'est sous LaTeX. Il ne nous reste alors plus qu'à demander à la fonction quapro de résoudre le problème.

Comme toutes les questions se ressemblent, nous nous contenterons de résoudre la dernière. Commençons par écrire un programme qui reçoit en entrée une matrice (par exemple la matrice de toutes les données, ou la matrice de toutes les données sauf quelques unes) ainsi que λ et qui rend *a* et *b* :

```
function [a,b]=svm(D,lambda)
//Séparation des données en deux sous-groupes
  s=size(D)
  n=s(1)
  groupe1=[]
  groupe2=[]
    for k=1:n
        if D(k,$)==2 then
            groupe1=[groupe1 ; D(k,2:$-1)]
        else
            groupe2=[groupe2 ; D(k,2:$-1)]
        end
    end
//Construction de Q,q,C,rhs,lb,ub.
  s1=size(groupe1)
  n1=s1(1)
  n2=n-n1
  Q=[2*eye(9,9) , zeros(9,n+1) ; zeros(1,n+10) ;
     zeros(n,10) 2*lambda*eye(n,n)]
  q=zeros(n+10,1)
  C=[groupe1 ones(n1,1) -eye(n1,n1) zeros(n1,n2) ;
     -groupe2 -ones(n2,1) -eye(n2,n2) zeros(n2,n1)]
  rhs=-ones(n,1)
  lb=[ -%inf*ones(10,1) ; zeros(n,1)]
  ub=[]
//Résolution du problème :
  x=quapro(Q,q,C,rhs,lb,ub)
  a=x(1:9)
  b=x(10)
endfunction
```

Notez au passage que si on a appelé la matrice des données Data on peut résoudre la première question en tapant dans Scilab (pour $\lambda = 1$) :

```
-->[a,b]=svm(Data,1)
```

Une fois le programme svm écrit, la validation croisée est très facile à mettre en place. On pourra par exemple écrire :

```
function res=valid_croisee(lambda)
  res=0
    for k=1:683
        D=Data
        D(k,:)=[]           //supprime la k-ième ligne
        [a,b]=svm(D,lambda)
        if (Data(k,$)==2 & Data(k,2:$-1)*a+b<0) then
            res=res+1
    elseif (Data(k,$)==4 & Data(k,2:$-1)*a+b>0) then
            res=res+1
        end
    end
  res=res/683*100
endfunction
```

Le temps d'exécution de ce programme est relativement long (environ quelques heures sur une machine récente). Le résultat est plutôt bon puisqu'on obtient 97% de points bien classés !

Deuxième partie

Applications

Chapitre 7

Vraisemblance Empirique

7.1 Introduction

Avant d'énoncer le théorème de Owen sur la vraisemblance empirique, rappelons quelques définitions et résultats classiques de processus empiriques.

Définition 7.1.1. Soit $(X_i)_{i\geq 1}$ une suite i.i.d. de variables aléatoires réelles. La loi de probabilité empirique de $(X_1, \ldots, X_n)$ est définie par : $\mathbb{P}_n = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \delta_{X_i}$.

Cette loi de probabilité met un poids de $1/n$ sur chacune des observations. Elle a de bonnes propriétés, par exemples celles énoncées par les théorèmes de Glivenko-Cantelli et Donsker.

Définition 7.1.2 (Vraisemblance). La vraisemblance d'une loi de probabilité $\mathbb{P}$ pour l'échantillon $(X_1, \ldots, X_n)$ est $L(\mathbb{P}) = \displaystyle\prod_{i=1}^{n} \mathbb{P}\{X_i\}$.

Proposition 7.1.3

La probabilité empirique de $(X_1, \ldots, X_n)$ est la loi de probabilité qui maximise la vraisemblance de l'échantillon.

7.2 Théorème de Owen

Le théorème de Owen part d'un constat simple, celui de la proposition 7.1.3 : la loi de probabilité empirique d'un échantillon maximise la vraisemblance de cet échantillon. Etant donné un échantillon, l'idée est alors de maximiser la vraisemblance de cet échantillon en ajoutant une contrainte.

Notation. Soit $(X_i)_{i\in\mathbb{N}}$ une suite de variables aléatoires réelles i.i.d. et $\mu \in \mathbb{R}$. On pose :

$$R(\mu) = \max\left\{\prod_{i=1}^{n} np_i \;,\; \sum_{i=1}^{n} p_i = 1 \;,\; p_i \geq 0 \;\sum_{i=1}^{n} p_i X_i = \mu\right\}.$$

Théorème 7.2.1 (Owen)

Soit $(X_i)_{i\in\mathbb{N}}$ une suite de variables aléatoires réelles i.i.d. et $\mu_0 = \mathbb{E}X_1$. Alors :

$$-2\log R(\mu_0) \xrightarrow{\mathcal{L}} \chi^2.$$

$R(\mu)$ étant défini par un problème de maximisation, il n'est pas étonnant que la démonstration du résultat de Owen fasse appel à la dualité Lagrangienne. Nous allons voir qu'il y a dualité forte dans ce problème, ce qui est utilisé dans la preuve du théorème mais qui n'est étrangement pas démontré !

Index