Séance de T.P. n°3

Cours : Optimisation
Optimisation sous contraintes

L'objectif de cette séance de travaux pratiques est de vous apprendre à coder des algorithmes généraux de minimisation avec contraintes : l'algorithme d'Uzawa et la méthode de pénalisation. L'algorithme d'Uzawa est rappelé au début du TP.

Partie I : Quelques rappels de cours

1 L'algorithme du gradient projeté

La méthode du gradient projeté s'inspire des méthodes usuelles de gradient. Supposons, d'une façon générale, que l'on souhaite minimiser une fonctionnelle $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ sur un ensemble de contraintes C. Si l'on construit une suite d'itérés de la forme $x^{k+1} = x^k + \rho_k d^k$, où d^k est une direction de descente, on ne peut pas être sûr que si x^k appartient à C, alors x^{k+1} appartiendra encore à C. Il faut donc « ramener » x^{k+1} dans C, ce que l'on fait en utilisant une projection.

Algorithme du gradient projeté

1. *Initialisation.*
   $k = 0$: on choisit $x^0 \in \mathbb{R}^n$ et $\rho_0 \in \mathbb{R}_+^*$.
2. *Itération k.*
   $$x^{k+1} = \Pi_C\left(x^k - \rho_k \nabla J(x^k)\right).$$
   Π_C désigne ici la projection sur C

Notons également le résultat de convergence :

Théorème 1 *On suppose que J est $\mathcal{C}^1$, de dérivée Lipschitzienne, et elliptique, c'est à dire qu'il existe $\alpha > 0$ tel que :*

$$\forall (x,y) \in (\mathbb{R}^n)^2,\ (\nabla J(x) - \nabla J(y), x - y) \geq \alpha.\|x-y\|^2.$$

Si l'on choisit le pas ρ_k dans un intervalle $[\beta_1, \beta_2]$ tel que $0 < \beta_1 < \beta_2 < \frac{2\alpha}{M}$, où α est la constante d'ellipticité de J et M, la constante de Lipschitz de la dérivée de J, alors la suite $(x^n)_{n\geq 0}$ d'itérés par la méthode du gradient projeté converge vers la solution du problème de minimisation.

2 L'algorithme d'Uzawa

Supposons que l'on souhaite résoudre sur $\mathbb{R}^n$ le problème $(\mathcal{P})\ \min J(x),\ x \in C$, où :

$$C = \{x \in \mathbb{R}^n, h(x) = 0,\ g(x) \leq 0\}, \text{ avec } h = (h_i)_{1\leq i\leq p} \text{ et } g = (g_j)_{1\leq j\leq q}.$$

Appelons $\mathcal{L}$, la fonction Lagrangienne associée à ce problème. Alors :

$$\forall x \in \mathbb{R}^n,\ \mathcal{L}(x,\lambda,\mu) = J(x) + (\lambda, h(x))_{\mathbb{R}^p} + (\mu, g(x))_{\mathbb{R}^q}\,.$$

On voit ici que la fonction Lagrangienne englobe à la fois la fonctionnelle J et les contraintes h et g. Elle représente donc bien le problème $(\mathcal{P})$. Avant de poursuivre, souvenons-nous de ce que l'on appelle **point selle**.

Définition 1 *On appelle* ***point selle*** *de $\mathcal{L}$ sur $\mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$, tout triplet (x^*, λ, μ) vérifiant l'équation :*

$$\mathcal{L}(x^*,\lambda,\mu) \leq \mathcal{L}(x^*,\lambda^*,\mu^*) \leq \mathcal{L}(x,\lambda^*,\mu^*),\ \forall (x,\lambda,\mu) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q. \tag{1}$$

Le théorème suivant nous permettra de comprendre *intuitivement* l'algorithme d'Uzawa. Pour une compréhension totale, il faudra s'intéresser au problème dual de $(\mathcal{P})$.

Théorème 2 *Supposons J, g et h convexes, de classe $\mathcal{C}^1$.*
Alors, le triplet $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$ est un point selle de $\mathcal{L}$ si, et seulement si ce triplet vérifie les conditions de Kuhn-Tucker.

Ce théorème nous aide à comprendre que, pour chercher le triplet $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$ vérifiant les conditions de Kuhn-Tucker, on peut procéder de la façon suivante :

- Pour $(\lambda^*, \mu^*) \in \mathbb{R}^p \times (\mathbb{R}_+)^q$, fixés, on peut chercher le minimum sans contrainte (i.e. sur tout l'espace $\mathbb{R}^n$) de la fonction $x \longmapsto \mathcal{L}(x, \lambda^*, \mu^*)$. Cela traduit le terme de droite de l'équation (1).
- Pour $x^* \in \mathbb{R}^n$ fixé, on cherche le maximum sur $\mathbb{R}^p \times (\mathbb{R}_+)^q$ de la fonction $(\lambda, \mu) \longmapsto \mathcal{L}(x^*, \lambda, \mu)$. C'est ce que traduit le terme de gauche de l'équation (1).

C'est cette idée qu'utilise l'algorithme d'Uzawa :

Algorithme d'Uzawa

1. *Initialisation.*
   $k = 0$: on choisit $\lambda^0 \in \mathbb{R}^p$ et $\mu^0 \in (\mathbb{R}_+)^q$.
2. *Itération k.*
   $\lambda^k = (\lambda_1^k, ..., \lambda_p^k) \in \mathbb{R}^p$ et $\mu^k = (\mu_1^k, ..., \mu_q^k) \in \mathbb{R}^q$ sont connus.
   (a) Calcul de $x^k \in \mathbb{R}^n$ solution de :

   $$(\mathcal{P}_k)\ \min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda^k, \mu^k).$$

   (b) Calcul de λ^{k+1} et μ^{k+1} par les formules :

   $$\begin{array}{rcl} \lambda_i^{k+1} & = & \lambda_i^k + \rho h_i(x^k),\ i \in \{1,...,p\} \\ \mu_j^{k+1} & = & \max(0, \mu_j^k + \rho g_j(x^k),\ j \in \{1,...,q\}. \end{array}$$

Enfin, signalons le théorème suivant qui pourrait se révéler utile en pratique :

Théorème 3 *On suppose que J est $\mathcal{C}^1$ et elliptique. Supposons de plus que h est affine, g est convexe de classe $\mathcal{C}^1$ et lipschitziennes. On suppose de plus que le Lagrangien $\mathcal{L}$ possède un point*

selle (x^*, λ^*, μ^*) sur $\mathbb{R}^n \times \mathbb{R}^p \times (\mathbb{R}_+)^q$. Alors, il existe ρ_1 et ρ_2, avec $0 < \rho_1 < \rho_2$ tels que, pour tout $\rho \in [\rho_1, \rho_2]$, la suite $(x^k)_{k\geq 0}$ générée par l'algorithme d'Uzawa converge vers x^*.
De plus, on sait que $\rho_2 = \frac{\alpha}{2(M_h^2+M_g^2)}$, avec α, la constante d'ellipticité de J, M_h et M_g, les constantes de Lipschitz associées à h et g.

Partie II : Exercices

EXERCICE N. 1

Optimisation d'un portefeuille d'actions

`TPUza1`

On considère le problème de l'Optimisation d'un portefeuille. Supposons que l'on possède n actions, que l'on représente par des variable aléatoires R_1, ..., R_n. Chaque action rapporte en moyenne à l'actionnaire $e_i = \mathbb{E}(R_i)$ (espérance de R_i) au bout d'un an. On suppose que l'on investit une somme S donnée, et l'on note $x_i \in \mathbb{R}$, la proportion de la somme investie dans l'action i. Ainsi, on a : $\displaystyle\sum_{i=1}^{n} x_i = 1$. Le portefeuille total est représenté par la variable aléatoire : $R = \displaystyle\sum_{i=1}^{n} x_i R_i$ et rapporte donc en moyenne : $\mathbb{E}(R) = \displaystyle\sum_{i=1}^{n} x_i e_i$.

On désire imposer un rendement donné $r_0 > 0$, ce qui se traduit par : $r_0 = \displaystyle\sum_{i=1}^{n} x_i e_i$.

On modélise le risque du portefeuille par : $\sigma^2(x) = \mathbb{E}[(R - \mathbb{R})^2]$.
On note $A = (a_{ij})_{\substack{1\leq i\leq n\\ 1\leq j\leq n}}$, la matrice de covariance définie par la relation :

$$\forall (i,j) \in \{1, ..., n\}^2,\ a_{ij} = \mathbb{E}[R_i - \mathbb{E}(R_i)]\mathbb{E}[R_j - \mathbb{E}(R_j)].$$

On peut alors écrire que $\sigma^2(x) = (x, Ax)$. On appelle J, la fonctionnelle définie sur $\mathbb{R}^n$ par :

$$J(x) = \frac{1}{2}(x, Ax).$$

On appelle également K, l'ensemble des contraintes : $K := \{x \in \mathbb{R}^n : (x, u) = 1 \text{ et } (x, e) = r_0\}$.
Le but de ce TP est de déterminer numériquement la solution du problème :

$$(\mathcal{P}) \begin{cases} \min J(x) \\ x \in K \end{cases}.$$

1. Mettre l'ensemble des contraintes sous la forme $K = \{x \in \mathbb{R}^n : Cx = f\}$, où C et f désignent respectivement une matrice et un vecteur à préciser. Rappeler comment se traduisent les conditions d'Optimalité de ce problème et formuler l'équation en x à résoudre à chaque itération.
2. On souhaite étudier un exemple concret. Supposons que : $\forall i \in \{1, ..., n\}$, $e_i = i$, que $r_0 = 2.5$. Pour les tests numériques, on se placera par exemple dans le cas où $n = 5$.
   Écrire un programme *genere.m* permettant de générer la matrice A à l'aide des instructions suivantes :
   ```
   A=diag(e./n);
   R=rand(n,n);
   A=A+0.1.*R'*R;
   ```
   Expliquer la dernière ligne du programme.

3. Pour différentes matrices A, programmer l'algorithme d'Uzawa. On n'oubliera pas d'imposer un nombre maximal d'itérations.
4. Quel inconvénient majeur constatez-vous ici ?
5. On appelle donc à présent $\tilde{K}$, l'ensemble défini par :

$$\tilde{K} = \{x \in \mathbb{R}^n : x \geq 0,\ (x,u) = 1 \text{ et } (x,e) = r_0\}.$$

Si l'on souhaite améliorer la résolution du problème, on est amené à résoudre : $\min\limits_{x \in \tilde{K}} J(x)$.
6. Comment doit-on choisir la constante ρ qui intervient dans l'algorithme d'Uzawa. Soyez précis.
7. Écrire l'algorithme d'Uzawa écrit sous sa forme la plus générale, puis le tester.

EXERCICE N. 2

Mise en œuvre d'une méthode de pénalisation

`TPPena`

On considère la fonctionnelle J définie sur $\mathbb{R}^2$ par : $J(x,y) = 2x^2 + 3xy + 2y^2$.
On appelle Q le cadrant défini par :

$$Q = \left\{x \leq -\frac{1}{2},\ y \leq -\frac{1}{2}\right\}$$

1. (a) Quelle est la solution du problème de minimisation de J sans contrainte ?
   (b) On appelle $X^\star$, le minimum de J sous la contrainte $X^\star \in Q$.
   Démontrer que, nécessairement, $\nabla J(X^\star) = 0$ ou $X^\star \in \partial Q$.
   (c) Mettre en œuvre la méthode du gradient projeté pour résoudre le problème de minimisation :
   $$\min_{(x,y) \in Q} J(x,y).$$
   Pensez-vous que la méthode du gradient conjugué peut être associée à la méthode de projection ? Pourquoi ?
   (d) Représenter les itérés par cette méhode.
2. **S'il vous reste du temps.** Nous allons reprendre le même problème que précédemment, et évaluer une méthode de pénalisation. On propose les étapes suivantes :
   (a) Mettre en place une fonction de pénalisation $x \mapsto \phi(x)$, en réfléchissant à l'expression qu'elle aura sur le cadrant Q. Déterminer alors le gradient de la fonctionnelle pénalisée.
   Remarque : attention au choix de la pénalisation ! La fonctionnelle pénalisée doit être différentiable.
   (b) Tracer les courbes de niveau de la nouvelle fonction coût et son gradient, pour plusieurs valeurs de ε. Que constatez-vous quant à la vitesse de variation de la fonction coût ? Dans la suite, on tracera le gradient uniquement sur le domaine admissible.
   (c) Pour une valeur petite de ε, par exemple $\varepsilon = 10^{-4}$ (pénalisation forte), et un point de départ $x_0 = (-0.3,\ 0.5)^t$, tester la méthode de pénalisation pour les méthodes de gradient à pas fixe et à pas optimal. En visualisant les itérés, peut-on dire que la vitesse de convergence est satisfaisante ? Répéter le test pour $\varepsilon = 0.5$ (pénalisation faible). Que peut-on dire de la convergence ? Quid de la solution trouvée ?
   (d) Déduire de ces observations une méthode qui converge plus rapidement.