

PROJET D'OPTIMISATION — Méthode de Markowitz

1 Un peu de contexte

Introduction à la Théorie Moderne de Portefeuille

La Théorie Moderne de Portefeuille (TMP) représente un pilier de la théorie financière, offrant un cadre systématique pour la construction de portefeuilles d'investissement optimaux. Développée par l'économiste Harry Markowitz dans les années 1950, la TMP a révolutionné la gestion de portefeuille en introduisant une rigueur mathématique dans sa construction. Cette introduction vise à élucider les fondements mathématiques des concepts clés de la TMP, permettant aux investisseurs de prendre des décisions éclairées dans le domaine complexe de la finance.

1. Frontière Efficiente : Au cœur de la TMP se trouve le concept de la frontière efficiente, qui représente l'ensemble des portefeuilles offrant les rendements attendus les plus élevés pour un niveau de risque donné ou le risque le plus faible pour un niveau de rendement donné. Mathématiquement, la frontière efficiente émerge comme la solution à un problème d'optimisation quadratique, équilibrant le compromis entre risque et rendement.

$$\max_{w \in \mathbb{R}^n} \mathbb{E}(R) - \lambda \cdot \mathbb{V}(R)$$

où $\mathbb{E}(R)$ représente le rendement attendu du portefeuille, $\mathbb{V}(R)$ représente la variance du rendement du portefeuille (le risque) et λ est le coefficient de préférence pour le risque.

2. Risque et Rendement : Au cœur de la TMP se trouve le concept de risque et de rendement. Le rendement attendu est calculé comme la moyenne pondérée des rendements individuels des actifs, tandis que le risque du portefeuille est quantifié par la variance ou l'écart-type des rendements du portefeuille.

$$\mathbb{E}(R) = \sum_{i=1}^n w_i \cdot \mathbb{E}(R_i) \quad \mathbb{V}(R) = \sum_{i=1}^n \sum_{j=1}^n w_i \cdot w_j \cdot \text{Cov}(R_i, R_j)$$

3. Diversification : La diversification, un principe clé de la TMP, désigne la stratégie de répartition des investissements entre différents actifs pour atténuer le risque du portefeuille. Mathématiquement, les avantages de la diversification se concrétisent par la réduction de la variance du portefeuille obtenue en incluant des actifs non corrélés ou négativement corrélés.

$$\sigma_p^2 = \sum_{i=1}^n \sum_{j=1}^n w_i \cdot w_j \cdot \sigma_i \cdot \sigma_j \cdot \rho_{ij}$$

4. Optimisation de Portefeuille : L'optimisation de portefeuille consiste à trouver l'allocation optimale des actifs pour maximiser le rendement attendu tout en minimisant le risque du portefeuille. L'optimisation de la moyenne-variance sert de cadre mathématique principal pour l'optimisation de portefeuille, résolvant des problèmes d'optimisation quadratique sous diverses contraintes.

$$\max_{w \in \mathbb{R}^n} \frac{\mathbb{E}(R) - R_f}{\sigma_p}$$

Le contexte du projet d'optimisation

Si l'on désigne par X la valeur (aléatoire) d'un portefeuille de n actifs et que la valeur de ce portefeuille varie peu autour de sa moyenne \bar{X} alors pour une fonction d'utilité (fonction concave et croissante) U régulière on peut écrire :

$$U(X) = U(\bar{X}) + U'(\bar{X})(X - \bar{X}) + \frac{1}{2}U''(\bar{X})(X - \bar{X})^2 + o((X - \bar{X})^2)$$

Si l'on prend maintenant l'espérance il vient :

$$\mathbb{E}[U(X)] = U(\bar{X}) + \frac{1}{2}U''(\bar{X})\mathbb{E}[(X - \bar{X})^2] + \mathbb{E}[o((X - \bar{X})^2)]$$

En introduisant $t \in \mathbb{R} \mapsto \rho(t) = -\frac{U''}{U'}(t)$ le coefficient d'aversion au risque on réécrit :

$$\mathbb{E}[U(X)] = U(\bar{X}) - U'(\bar{X})\frac{\rho(\bar{X})}{2}\mathbb{E}[(X - \bar{X})^2] + \mathbb{E}[o((X - \bar{X})^2)]$$

On peut écrire (en négligeant le reste d'ordre 2) :

$$\mathbb{E}[U(X)] \sim U\left(\bar{X} - \frac{1}{2}\rho\mathbb{E}[(X - \bar{X})^2]\right)$$

On définit alors la quantité $E_c(X) := \bar{X} - \frac{\rho(\bar{X})}{2}\mathbb{E}[(X - \bar{X})^2]$, équivalent certain du portefeuille à échéance. L'approche de Markowitz consiste à maximiser cet équivalent certain. Si X_0 désigne la valeur initiale du portefeuille, on écrit :

$$E_c(x) = X_0(1 + \langle \bar{\mu}, x \rangle) - \frac{\rho(X_0(1 + \langle \bar{\mu}, x \rangle))}{2}X_0^2\langle \bar{K}x, x \rangle$$

où $\bar{K} = \mathbb{E}[(\mu - \bar{\mu})(\mu - \bar{\mu})^T]$ avec μ performance (aléatoire) du portefeuille et $\bar{\mu}$ sa performance moyenne. On a alors que \bar{K} représente la matrice de variance-covariance de la performance.

Si l'aversion au risque est constante on est amené à maximiser

$$J(x) = \langle \bar{\mu}, x \rangle - \frac{\rho}{2}X_0\langle \bar{K}x, x \rangle.$$

Les vecteurs $\bar{\mu}$ et la matrice \bar{K} sont dans cette approche des prévisions.

Enjeux du projet

A partir du problème d'optimisation précédent, le but de ce projet est d'abord de prouver l'existence d'une solution (unique ou non) qui vérifie :

$$\min_{s \in U_{ad}} \max_{x \in X_{ad}} J(\mu, x) = \max_{x \in X_{ad}} \min_{s \in U_{ad}} J(\mu, x).$$

où U_{ad} représente l'ensemble des performances aléatoires μ du portefeuille et X_{ad} représente l'ensemble des valeurs aléatoires d'un portefeuille de n actifs.

Une fois avoir prouvé l'existence de ce minmax et qu'il soit bien atteint, l'objectif est de pouvoir l'implémenter numériquement pour obtenir rapidement les meilleurs placements possibles du portefeuille. Cette implémentation se fera avec l'algorithme de Frank et Wolfe grâce à une recherche linéaire par algorithme du simplexe. Dans le cadre de ce projet pour simplifier l'étude, on considère que la matrice K est donnée (symétrique, définie, positive) et l'on notera $J(s, x) =: J(\mu, x)$, $S_{ad} = U_{ad} \times K$, K représentant la matrice de variance-covariance de la performance. On considérera que la performance est constante. On suppose également, sans perte de généralité, que $X_0 = 1$.

2 Vers une approche multi-scenario

Question 1

Justifier que si X_{ad} et U_{ad} sont convexes et compacts le problème (\mathcal{P}) admet une solution et

$$\min_{s \in U_{ad}} \max_{x \in X_{ad}} J(\mu, x) = \max_{x \in X_{ad}} \min_{s \in U_{ad}} J(\mu, x).$$

Que peut-on dire de l'unicité?

Solution

Théorème 2.1 : *Soit $E \subset \mathbb{R}^n$ un sous-ensemble de l'espace euclidien. Alors E est compact si et seulement si E est fermé et borné.*

Théorème 2.2 : *Soit X un espace métrique compact et $f : X \rightarrow \mathbb{R}$ une fonction continue. Alors, il existe un $x \in X$ tel que $f(x)$ soit le maximum de f sur X , c'est-à-dire que $f(x) \geq f(y)$ pour tout $y \in X$.*

Commençons par noter que $X_{ad} \subset \mathbb{R}^n$ et $U_{ad} \subset \mathbb{R}^n$. D'après le Théorème 2.1, U_{ad} et X_{ad} sont des compacts, donc ce sont des fermés bornés de \mathbb{R}^n . De plus, par continuité du produit scalaire, J est continue en x et en μ sur $U_{ad} \times X_{ad}$. Donc J est bornée sur $U_{ad} \times X_{ad}$ car ces deux ensembles sont bornés dans $(\mathbb{R}^n)^2$.

Étape 1 : Existence d'un maximum pour chaque μ :

Pour chaque μ fixé, considérons le problème d'optimisation $\max_{x \in X_{ad}} J(\mu, x)$. Comme $J(\mu, x)$ est continue et X_{ad} est compact, d'après le Théorème 2.2, le maximum de $J(\mu, x)$ sur $x \in X_{ad}$ existe et est atteint en un point $\hat{x}(\mu)$.

Étape 2 : Sélection du meilleur μ dans U_{ad} :

Considérons maintenant l'ensemble des valeurs $J(\mu, \hat{x}(\mu)) : \mu \in U_{ad}$. Comme U_{ad} est compact et que $J(\mu, \hat{x}(\mu))$ est continue, d'après le Théorème 2.2, cet ensemble admet un minimum et ce minimum est atteint. On notera ce minimum c^* .

Étape 3 : Trouver μ correspondant :

Soit $\hat{\mu}$ un point de U_{ad} tel que $J(\hat{\mu}, \hat{x}(\hat{\mu})) = c^*$. Un tel point existe car c^* est le minimum de l'ensemble $J(\mu, \hat{x}(\mu)) : \mu \in U_{ad}$ et $J(\mu, \hat{x}(\mu))$ est continue.

Conclusion :

Le couple $(\hat{\mu}, \hat{x}(\hat{\mu}))$ est une solution au problème $\min_{\mu \in U_{ad}} \max_{x \in X_{ad}} J(\mu, x)$ car $J(\hat{\mu}, \hat{x}(\hat{\mu}))$ atteint le minimum de $\max_{x \in X_{ad}} J(\mu, x) : \mu \in U_{ad}$.

Égalité min max et max min :

Théorème 2.3 : Soient U_{ad} et X_{ad} des espaces convexes et compacts et J une fonction continue sur $X_{ad} \times U_{ad}$, *quasi-concave-convexe* et *semi-continue supérieurement*, *semi-continue inférieurement*. Alors $\sup_{x \in X_{ad}} \inf_{\mu \in U_{ad}} J = \inf_{\mu \in U_{ad}} \sup_{x \in X_{ad}} J$.

Théorème 2.4 : Soit Ω un ouvert convexe de \mathbb{R}^d . Soit $J : \Omega \rightarrow \mathbb{R}$ une application de Ω dans \mathbb{R} .

- a) Si J est de classe \mathcal{C}^1 sur Ω , J est convexe si et seulement si, pour tous x et y dans Ω , $J(y) \geq J(x) + dJ(x)(y - x)$.
- b) Si J est de classe \mathcal{C}^1 sur Ω , J est convexe si et seulement si, pour tous x et y dans Ω , $(dJ(y) - dJ(x)) \cdot (y - x) \geq 0$.
- c) Si J est de classe \mathcal{C}^2 sur Ω , J est convexe si et seulement si, pour tout x dans Ω , $d^2J(x)$ est une forme bilinéaire semi-définie positive.
- d) Si J est de classe \mathcal{C}^2 sur Ω et si, pour tout x dans Ω , $d^2J(x)$ est une forme bilinéaire définie positive, alors J est strictement convexe.

On a :

$$\frac{\partial^2 J}{\partial \mu^2}(\mu, x) = 0 \quad \frac{\partial^2 J}{\partial x^2}(\mu, x) = -\rho K$$

Par le Théorème 2.4, on obtient que $J(\mu, x)$ est convexe par rapport à μ , car son gradient est affine, et strictement concave par rapport à x , car $\rho > 0$ et K est symétrique définie positive. De plus, J est continue par rapport à ses variables. En appliquant le Théorème 2.3, nous obtenons l'égalité souhaitée.

Unicité :

Si $J(\mu, x)$ est strictement convexe ou strictement concave par rapport aux deux variables, alors la solution est unique. D'après la question précédente, $J(\mu, x)$ est strictement concave par rapport à x , et $J(\mu, x)$ est convexe par rapport à μ si et seulement si $\mu \neq 0$.

Question 2

En déduire que l'on pourra dans ce cadre résoudre le problème en recherchant un point selle de J .

Solution

Dans notre contexte, U_{ad} et X_{ad} sont des ensembles convexes et compacts, et $J(\mu, x)$ est une fonction continue.

Un point selle de $J(\mu, x)$ est un point où la fonction atteint à la fois un maximum sur x et un minimum sur μ . En d'autres termes, pour résoudre le problème (\mathcal{P}) , nous cherchons à trouver un point (s^*, x^*) tel que :

$$\forall s \in U_{ad} \quad \forall x \in X_{ad} \quad J(\mu^*, x) \leq J(\mu^*, x^*) \leq J(\mu, x^*)$$

Autrement dit, trouver un point selle revient à résoudre (\mathcal{P}) grâce à la structure convexe de $J(\mu, x)$ par rapport à s et la structure concave de $J(\mu, x)$ par rapport à x .

Remarque : Ce point selle peut ne pas être unique, cela dépend de J . On a la l'unicité du point si et seulement si $J(\mu, x)$ est strictement convexe par rapport à s et strictement concave par rapport à x .

3 Vers une résolution algorithmique

3.1 Algorithme de Franck et Wolfe sous contraintes

On va utiliser à μ fixé, l'algorithme de Franck et Wolfe pour résoudre $\max_{x \in X_{ad}} J(\mu, x)$. Cet algorithme converge dès lors que la fonction est C^1 et l'ensemble des contraintes borné.

1. Initialisation : on se donne $x_0 \in X_{ad}$.
2. Recherche linéaire par algorithme du simplexe : si l'on écrit

$$J(x) = J(x_k) - \langle \nabla J(x_k), x - x_k \rangle + o(x - x_k),$$

maximiser J revient localement à maximiser $\langle \nabla J(x_k), x \rangle$ pour $x \in X_{ad}$ ce qui peut-être fait via l'algorithme du simplexe. Notons \tilde{x}_{k+1} la solution de ce problème linéaire. On définit alors

$$J(x) = J(x_k) - \langle \nabla J(x_k), x - x_k \rangle + o(x - x_k),$$

On justifiera que pour J comme ci-dessus, τ_{k+1} est bien défini de façon unique et l'on en donnera une expression analytique (ce qui évitera une étape de calcul dans l'algorithme).

3. Itérer tant que $\|x_{k+1} - x_k\| \geq \varepsilon \|x_0 - x_1\|$ pour $\varepsilon > 0$ fixé.

Solution

On note $J = -\tilde{J}$, \tilde{J} est convexe et le problème de maximisation devient un problème de minimisation

$$\begin{aligned} \tilde{J}(x_{k+1}) &= \tilde{J}((1 - \tau_{k+1})x_k + \tau_{k+1}\tilde{x}_{k+1}) \\ &\leq J(x_k) + \tau_{k+1} \langle \nabla J(x_k), x_{k+1} - x_k \rangle + \frac{c}{2}\tau_{k+1}^2 \quad (\text{Constante de convexité}) \\ &\leq J(x_k) + \tau_{k+1} \langle \nabla J(x_k), \tilde{x}_{k+1} - x_k \rangle + \frac{c}{2}\tau_{k+1}^2 \quad (\text{Car } \tilde{x}_{k+1} \text{ solution du Problème}) \\ &\leq (1 - \tau_{k+1})J(x_k) + \tau_{k+1}[J(x_k) + \langle \nabla J(x_k), \tilde{x}_{k+1} - x_k \rangle] + \frac{c}{2}\tau_{k+1}^2 \\ &\leq (1 - \tau_{k+1})J(x_k) + \tau_{k+1}J(\tilde{x}_{k+1}) + \frac{c}{2}\tau_{k+1}^2 \\ J(x_{k+1}) - J(\tilde{x}_{k+1}) &\leq (1 - \tau_{k+1})J(x_k) - (1 - \tau_{k+1})J(\tilde{x}_{k+1}) + \frac{c}{2}\tau_{k+1}^2 \\ &\leq (1 - \tau_{k+1})(J(x_k) - J(\tilde{x}_{k+1})) + \frac{c}{2}\tau_{k+1}^2 \\ \Delta_{k+1} &\leq (1 - \tau_{k+1})\Delta_k + \frac{c}{2}\tau_{k+1}^2 \end{aligned}$$

en choisissant $\tau_k = \frac{2}{2+k}$ (qui garantie la convergence et l'utilisation d'un pas indépendant des paramètres, c'est-à-dire ne nécessitant aucun paramètre de fonction ou d'estimation de paramètre) on a :

$$\Delta_k \leq \frac{C}{k+2} \xrightarrow{k \rightarrow \infty} 0 \quad (\text{On est dans un ensemble compact et donc } C)$$

3.2 Introduction de la fonction duale et algorithme complet

On introduit :

$$\forall \mu \in \mathbb{R}^n \quad H(\mu) = \max_{x \in X_{ad}} (\langle \mu, x \rangle - \frac{\rho}{2} \langle Kx, x \rangle).$$

Question 1

Justifier que l'on peut écrire $\rho K = LL^T$ où L est une matrice triangulaire inférieure. Montrer que

$$H(\mu) = \frac{1}{2} (\langle K\mu, \mu \rangle - d^2(L^{-1}\mu, L^T X_{ad})),$$

où pour un convexe fermé C on note $d^2(y, C) = \|y - \Pi_C(y)\|^2 =: \varphi_C(y)$ la distance de y à C .

Solution

Théorème 2.1 : *Soit A une matrice symétrique définie positive. Alors il existe une matrice triangulaire inférieure L telle que $A = LL^T$. De plus, si on impose aux coefficients diagonaux de L d'être strictement positifs, cette factorisation est unique.*

Dans la Partie 2, on a considéré que la matrice K est donnée symétrique définie positive (SDP) et φ est un scalaire, ie φK reste SDP. On applique alors le théorème de décomposition de Cholesky. Alors il existe une matrice réelle triangulaire inférieur L tel que :

$$\varphi K = LL^T$$

Obtention du x^* :

Soit $\forall x \in \mathbb{R}^n \quad \Phi(x) = \frac{1}{2} \langle Ax, x \rangle - \langle b, x \rangle$, avec A une matrice symétrique définie positive et b un vecteur de \mathbb{R}^n .

$$\begin{aligned} \Phi(x+h) &= \frac{1}{2} \langle Ax + Ah, x+h \rangle - \langle b, x+h \rangle \\ &= \frac{1}{2} [\langle Ax, x \rangle + \langle Ax, h \rangle + \langle Ah, x \rangle + \langle Ah, h \rangle] - \langle b, x \rangle - \langle b, h \rangle \\ &= \Phi(x) + \langle \frac{Ax}{2} + \frac{A^t x}{2} - b, h \rangle + \frac{1}{2} \langle Ah, h \rangle \end{aligned}$$

On a alors $\nabla \Phi(x) = \frac{Ax}{2} + \frac{A^t x}{2} - b$, en résolvant le problème d'optimisation $\min_{x \in \mathbb{R}^n} \Phi(x)$, on a l'égalité :

$$\nabla \Phi(x) = 0 \Leftrightarrow Ax = b$$

D'où le résultat suivant : x solution de $Ax = b \Leftrightarrow x$ minimum de $\Phi(x)$

En modifiant avec les données de notre problème, on a $A = \rho K = LL^T$ et $b = \mu$:

$$H(\mu) = \max_{x \in X_{ad}} (\langle \mu, x \rangle - \frac{\rho}{2} \langle Kx, x \rangle) = \min_{x \in X_{ad}} (\frac{\rho}{2} \langle Kx, x \rangle - \langle \mu, x \rangle)$$

En appliquant l'égalité $\varphi K = LL^T$ à $H(\mu)$ et $x^* = (LL^T)^{-1}\mu$, on obtient :

$$H(\mu) = \langle \mu, (L^T)^{-1} L^{-1} \mu \rangle - \frac{1}{2} \langle \mu, (L^T)^{-1} L^{-1} \mu \rangle = \frac{1}{2} \langle \mu, (L^T)^{-1} L^{-1} \mu \rangle = \frac{1}{2} \langle L^{-1} \mu, L^{-1} \mu \rangle = \frac{1}{2} \|L^{-1} \mu\|^2$$

Comme $L^{-1}\mu$ est orthogonale à $L^{-1}\mu - \Pi_{L^T X_{ad}}(L^{-1}\mu)$, on a l'égalité suivante :

$$\|L^{-1}\mu\|^2 + \|L^{-1}\mu - \Pi_{L^T X_{ad}}(L^{-1}\mu)\|^2 = \|L^T \mu\|^2$$

Ce qui donne dans $H(\mu)$:

$$\begin{aligned}
H(\mu) &= \frac{1}{2}(\|L^T \mu\|^2 - \|L^{-1} \mu - \Pi_{L^T X_{ad}}(L^{-1} \mu)\|^2) \\
&= \frac{1}{2}(\langle L^T \mu, L^T \mu \rangle - d^2(L^{-1} \mu, L^T X_{ad})) \\
&= \frac{1}{2}(\langle LL^T \mu, \mu \rangle - d^2(L^{-1} \mu, L^T X_{ad})) \\
&= \frac{1}{2}(\langle K \mu, \mu \rangle - d^2(L^{-1} \mu, L^T X_{ad}))
\end{aligned}$$

Question a

Justifier que la fonction φ_C est bien définie.

Solution

Nous devons vérifier deux choses :

1. que $\forall y \in \mathbb{R}^n \quad \Pi_C(y)$ existe
 2. que la norme $\forall y \in \mathbb{R}^n \quad \|y - \Pi_C(y)\|^2$ est finie
1. On a la proposition suivante : $\forall y \in \mathbb{R}^n$ la projection $\Pi_C(y)$ existe $\Leftrightarrow C$ est un ensemble convexe et fermé. Comme C vérifie cette condition. Alors $\forall y \in \mathbb{R}^n \quad \exists! \Pi_C(y) \in C$ tel que $\|y - \Pi_C(y)\|$ est minimale.
2. Puisque C est convexe fermé, la distance $\|y - \Pi_C(y)\|$ est bien définie et non négative donc $\|y - \Pi_C(y)\|^2$ l'est aussi.

Ainsi nous avons montré que $\forall y \in \mathbb{R}^n \quad \varphi_C(y) = \|y - \Pi_C(y)\|^2$ est bien définie, car $\Pi_C(y)$ existe pour tout y et la norme $\|y - \Pi_C(y)\|^2$ est finie.

Question b

Montrer que φ_C est $C^1(\mathbb{R}^n)$ et $\nabla \varphi(y) = 2(y - \Pi_C(y))$

Solution

Nous devons vérifier deux choses :

1. que la fonction φ_C est différentiable sur \mathbb{R}^n
 2. que le gradient de φ_C est continue sur \mathbb{R}^n
1. $\varphi_C(y) = \|y - \Pi_C(y)\|^2$ est différentiable sur \mathbb{R}^n par composition de fonctions différentiable sur \mathbb{R}^n (la norme euclidienne et la projection sont différentiable).
2. On a les égalités suivantes :

$$\begin{aligned}
\nabla \varphi(y) &= \nabla(\|y - \Pi_C(y)\|^2) \\
&= 2(y - \Pi_C(y)) \cdot \nabla(y - \Pi_C(y)) \\
&= 2(y - \Pi_C(y))(I_d - P_C) \\
&= 2(y - \Pi_C(y)) - 2(P_C(y) - P_C \Pi_C(y)) \\
&= 2(y - \Pi_C(y))
\end{aligned}$$

car $2(P_C(y) - P_C\Pi_C(y)) = 0$. De plus le gradient est continue par continuité de fonction polynomiale.

De **1** et **2**, on obtient que $\varphi(y)$ est C^1 .

Question c

En déduire que H est C^1 sur \mathbb{R}^n et montrer que $\nabla H(\mu) = x(\mu)$ où $x(\mu) \in X_{ad}$ est le point qui réalise le maximum de $J(\mu, x)$.

Solution

D'après la Question b, on a que $\forall y \in \mathbb{R}^n \varphi_C(y)$ est C^1 , il reste juste à montrer que $g(\mu) = \langle K\mu, \mu \rangle$ est C^1 .

- $g(\mu)$ est différentiable car $\nabla \langle K\mu, \mu \rangle = 2K\mu$
- $\nabla g(\mu)$ est continue car $2K\mu$ est une transformation linéaire

Donc $g(\mu)$ est C^1 .

Ainsi $H(\mu)$ est C^1 par combinaison linéaire entre deux fonction C^1 (g et φ_C).

$$\begin{aligned}\nabla H(\mu) &= \frac{1}{2}(2K\mu - 2(L^{-1}\mu - \Pi_{L^T X_{ad}}(L^{-1}\mu))) \\ &= K\mu - L^{-1}\mu + \Pi_{L^T X_{ad}}(L^{-1}\mu) \\ &= K\mu - L^{-1}\mu - LL^T\mu + L^{-1} + (LL^T)^{-1}\mu \\ &= (LL^T)^{-1}\mu \\ &= x(\mu)\end{aligned}$$

où $x(\mu)$ est solution du système linéaire :

$$\begin{aligned}LL^T x = \mu &\Leftrightarrow \min_{x \in X_{ad}} J(\mu, x) = \frac{1}{2}\langle LL^T x, x \rangle - \langle \mu, x \rangle \\ &\Leftrightarrow \max_{x \in X_{ad}} J(\mu, x) = \langle \mu, x \rangle - \frac{1}{2}\langle LL^T x, x \rangle\end{aligned}$$

Question 2

Justifier que H est convexe sur \mathbb{R}^n . Quand est-elle strictement convexe ?

Solution

$Hess(H(\mu)) = (LL^T)^{-1}$ et comme L est une matrice triangulaire inférieure alors ses valeurs propres sont les éléments de la diagonale principale, et comme LL^T est SDP alors $(LL^T)^{-1}$ est aussi SDP et donc $Hess(H(\mu))$ est SDP.

Donc H est convexe sur \mathbb{R}^n , H est strictement convexe si et seulement si $(LL^T)^{-1}$ est défini positif ce qui est le cas.

Ainsi H est strictement convexe.

Question 3

Mettre en oeuvre l'algorithme suivant : on cherche à minimiser la fonction duale.

- Initialisation : pour U_{ad} donné on se donne $\mu_0 \in U_{ad}$.
- Itération : à μ_k connu on cherche $\nabla H(\mu_k) = x(\mu_k)$. Une fois ce gradient à disposition on va chercher à minimiser (en μ) dans cette direction (par simplexe comme dans l'algorithme de Franck et Wolfe). Cette étape fournit un point μ_{k+1} . On pose ensuite :

$$\mu_{k+1} := (1 - \rho_{k+1})\mu_k + \rho_{k+1}\tilde{x}_{k+1}, \quad \rho_{k+1} := \arg \max_{\rho \in [0,1]} H((1 - \rho)\mu_k + \rho\tilde{x}_{k+1}).$$

- Arrêt dès lors que : $\|\mu_{k+1} - \mu_k\| < \varepsilon_k \|\mu_1 - \mu_0\|$.

Solution

```
import numpy as np

# Définition de J
def J(mu, x, rho, K):
    return np.dot(mu, x) - 0.5 * np.dot(np.dot(rho * K, x), x)

# Définition du gradient de J par rapport à x
def gradient_J(mu, x, rho, K):
    return mu - rho * np.dot(K, x)

# Fonction pour maximisé J en fonction de x
def max_J_en_x(mu, rho, K, epsilon, N=1e6):
    x_k = np.random.rand(len(K))
    k = 0
    x_k_prev = np.zeros_like(x_k) # Vecteur de 0 de même taille que x_0 fournit

    # Condition de sortie
    while np.linalg.norm(x_k - x_k_prev) >= epsilon * np.linalg.norm(x_k_prev):

        # Utilisation du simplexe
        x_tilde_k_1 = np.zeros_like(x_k)
        max_lambda = np.argmax(gradient_J(mu, x_k, rho, K))
        x_tilde_k_1[max_lambda] = 1

        pas_opti = 2 / (2 + k) # formule du pas optimisé
        x_k = (1 - pas_opti) * x_k + pas_opti * x_tilde_k_1

        k += 1
        x_k_prev = x_k.copy()

    if k > N: # Pour éviter une boucle infini
        break

    return x_k

# Fonction pour minimisé H en fonction de mu
def min_H_en_mu(K, rho, epsilon, N=1e6):
    mu_k = np.random.rand(len(K))
```

```

k = 0
mu_k_prev = np.zeros_like(mu_k) # Vecteur de 0 de même taille que mu_0 fournit

# Condition de sortie
while np.linalg.norm(mu_k - mu_k_prev) < epsilon * np.linalg.norm(mu_k_prev):
    grad_H = max\_J_en_x(mu_k, rho, K, epsilon)

    # Utilisation du simplexe
    mu_tilde_k_1 = np.zeros_like(mu_k)
    max\_index = np.argmin(grad_H)
    mu_tilde_k_1[max\_index] = 1

    pas_opti = 2 / (2 + k) # formule du pas optimisé
    mu_k = (1 - pas_opti) * mu_k + pas_opti * mu_tilde_k_1

    k += 1
    mu_k_prev = mu_k.copy()

    if k > N: # Pour éviter une boucle infini
        break

return mu_k

# Exemple d'utilisation
n = 5
mu = np.random.rand(n) # On crée un vecteur mu aléatoire

# On crée une matrice symétrique définie positive K
def SDP_matrix(epsilon=1e-8):
    # On crée une matrice de taille n x n aléatoire
    matrix = np.random.rand(n,n)

    # On rend la matrice K symétrique
    matrix = np.dot(matrix, matrix.T)

    # On calcule les valeurs propres et les vecteurs propres
    eigenvalues, eigenvectors = np.linalg.eigh(matrix)

    # On rend les valeurs propres non négatives
    eigenvalues[eigenvalues < 0] = epsilon

    # On reconstruit la matrice avec les vp et vect propre
    positive_definite_matrix = np.dot(eigenvectors, np.dot(np.diag(eigenvalues), eigenvectors.T))
    return positive_definite_matrix

K=SDP_matrix()

rho = 1
epsilon = 1e-6

x_solution = max\_J_en_x(mu, rho, K, epsilon)

```

```

print("x maximisé:", x_solution)

mu_solution = min_H_en_mu(K, rho, epsilon)
print("mu minimisé:", mu_solution)

# Les solutions renvoyées sont :
x maximisé: [0. 1. 0. 0. 0.]
mu minimisé: [0.13545492 0.61709843 0.95097715 0.76610093 0.80776526]

```

Explication du Code :

On définit la fonction J et le gradient de J par rapport à x .

Maximisation de J par rapport à x avec l'algorithme de Frank-Wolfe :

- La fonction `max_J_en_x` prend en entrée μ , K , et un seuil de convergence ϵ .
- Elle initialise x_k aléatoirement, puis itère jusqu'à ce que la différence entre les valeurs de x_k à deux itérations successives soit inférieure à ϵ .
- À chaque itération, elle choisit une direction de descente en utilisant le gradient de J , puis calcule la taille du pas optimal en utilisant la formule $\frac{2}{2+k}$.
- Elle met à jour x_k en fonction de la taille du pas et répète le processus jusqu'à convergence.
- Elle renvoie le vecteur x qui maximise la fonction J .

Minimisation de H par rapport à μ avec l'algorithme de Frank-Wolfe :

- La fonction `min_H_en_mu` prend en entrée K , ρ , et un seuil de convergence ϵ .
- Elle initialise μ_k aléatoirement, puis itère jusqu'à ce que la différence entre les valeurs de μ_k à deux itérations successives soit inférieure à ϵ .
- À chaque itération, elle utilise la fonction `max_J_en_x` pour obtenir le gradient de H par rapport à μ , puis calcule la taille du pas optimal en utilisant la formule $\frac{2}{2+k}$.
- Elle met à jour μ_k en fonction de la taille du pas et répète le processus jusqu'à convergence.
- Elle renvoie le vecteur μ qui minimise la fonction H .

Exemple d'utilisation :

- Nous générons un vecteur aléatoire μ et une matrice symétrique définie positive K .
- En utilisant les fonctions `max_J_en_x` et `min_H_en_mu`, nous maximisons $J(\mu, x)$ par rapport à x et minimisons $H(\mu)$ par rapport à μ .
- Nous imprimons ensuite les solutions obtenues.

Submitted by ROHIMUN Shakil on 9 Mars 2024.