Apprentissage de connexions dans les données financières Application à de l'optimisation de portefeuille d'actions sur données CAC40

Dylan Fagot

6 mai 2023

Table des matières

1 Introduction

Ce projet d'analyse de données financières se base sur une publication du MIT [1], proposant un modèle permettant de quantifier les connexions entre les rendements de différents cours et la tendance globale.

2 Diversification de portefeuilles

Cette section présente la stratégie de diversification présentée dans la publication [1].

2.1 Minimiser les liens

Le scénario à éviter est celui de la chute de la valeur de l'ensemble du portefeuille. Il faut évidement éviter de tout investir sur une action, car une chute du cours impacterait l'ensemble du portefeuille. Cela demande donc de diversifier en investissant sur plusieurs actions. Toutefois, le fait de multiplier le nombre de cours n'est pas forcément une stratégie gagnante. En effet, il peut s'avérer que ces cours évoluent de façon similaire.
La clé d'une diversification de portefeuille réussie est donc d'investir sur des cours présentant le moins de corrélation possible.

2.2 La matrice de covariance, une première approche

Une méthode de diversification présentée dans [1] se base sur la matrice de covariance entre les différents cours. En minimisant la covariance d'un ensemble pondéré d'actions formant un portefeuille, il est possible d'obtenir la répartition optimale de possession. Comme expliqué dans la publication [2], la covariance capture les interdépendances entre deux cours, mais contient également du liens avec les autres cours. Pour cette raison, la stratégie de minimisation de covariance n'est pas la plus adaptée pour faire de l'optimisation de portefeuille.

2.3 La matrice de connexion

La publication utilisée dans le cadre de ce projet [] présente un modèle permettant de capturer les liens que possède chaque cours avec chacun des autres cours, et le cours du marché.

3 Concepts mathématiques

Cette section présente les différents concepts sur lesquels se basent ce projet. Ceux-ci sont issus du domaine des statistiques, du machine learning et de l'optimisation mathématique.

3.1 Interpréter les covariances

La mesure de covariance entre deux variables aléatoires x et y sur N points se calcule via la formule :

$$cov_{1,2} = \frac{1}{N-1}\sum_{n=1}^{N}(x_1 - \bar{x_1})(x_2 - \bar{x_2}) \tag{1}$$

Lorsque cette valeur est élevée, cela signifie que les deux variables sont corrélées : elles tendent à évoluer dans le même sens (si covariance positive) ou en sens opposés (si covariance négative). Si cette valeur est proche de zéro, les deux variables sont décorrélées : leurs variations ne coincident pas. Cela est illustré dans la figure suivante.
Il semble donc intuitif d'utiliser cette métrique pour identifier des dépendances, afin de maximiser la diversité en achetant des actions qui sont trop corrélées.
Le calcul de covariance sur plusieurs cours peut se réaliser matriciellement sous la forme $\mathbf{Q} = \frac{1}{n}\sum_{n=1}^{N}[x_1, ..., x_C]^\top [x_1, ..., x_C]$ avec C le nombre de cours. Les termes diagonaux de cette matrice sont les variances de chaque variables, tandis que les termes extradiagonaux sont les covariances entre chaque couple de variables.

$$\mathbf{Q} = \begin{pmatrix} var_1 & \dots & cov_{1,C} \\ \vdots & \ddots & \vdots \\ cov_{1,C} & \dots & var_C \end{pmatrix} \tag{2}$$

3.2 Maximiser la diversification

Comme rappelé dans la publication [1], le problème de diversification de portefeuille peut être posé sous la forme d'un problème d'optimisation mathématique de minimisation de la covariance. Cette section explique comment traiter ce problème d'optimisation multi-objectif.

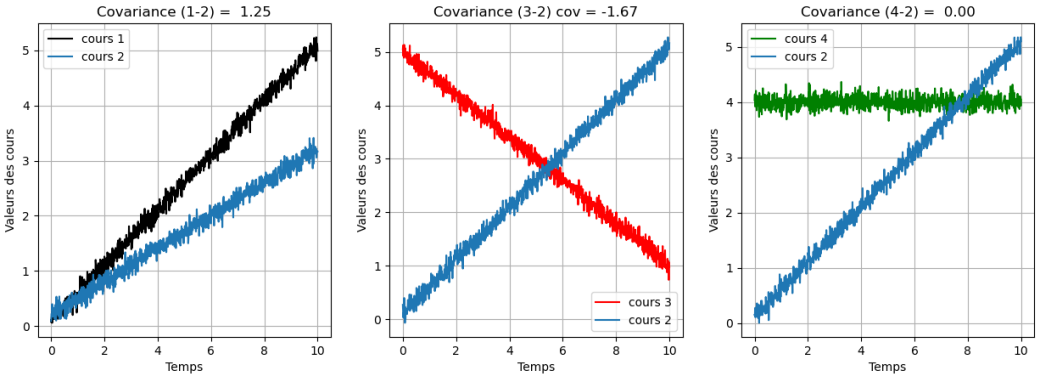

FIGURE 1 – Illustration de trois cas de covariance entre des cours 1 et 2 (gauche), 3 et 2 (centre), 4 et 2 (droite). Le cours 4 (vert) apparait comme décorrelé des cours 1, 2 et 3.

3.2.1 Minimiser la covariance

Nous notons dans ce projet le nombre de cours étudiés C. En notant $w = [w_1, ..., w_C]^\top$ le vecteur contenant les coefficients positifs de répartition des actions, et $\mathbf{Q}$ la matrice de covariance de leurs cours, le problème de minimisation de covariance s'écrit :

$$w_{opt} = \underset{w}{\operatorname{argmin}}\ w^\top \mathbf{Q} w \tag{3}$$

Sous cette forme, le problème présente une solution triviale $w = [0, ..., 0]^\top$. Pour l'éviter, il est possible de poser une contrainte sur les valeurs de w :

$$\sum_{c=1}^{C} w_c = 1 \tag{4}$$

De par la positivté des termes de w, cela revient à fixer la contrainte $||w||_1 = 1$. Les valeurs de w peuvent alors être interprétées comme des pourcentages. Si $w_{opt} = [0.3\ ,\ 0.7]^\top$, cela signifie que le portefeuille doit être composé à 30% d'action 1 et à 70% d'actions 2.

3.2.2 Exiger du rendement

Par ailleurs, un autre objectif de la construction de portefeuille est que celui-ci soit rentable. En notant $\bar{r} = [\bar{r}_1, ..., \bar{r}_C]^\top$ le vecteur des rendements moyens de chacune des actions, le rendement moyen global du portefeuille $\bar{r}_p$ s'écrit :

$$\bar{r}_p = w^\top \bar{r} = \sum_{c=1}^{C} w_c \bar{r}_c \tag{5}$$

La contrainte de rentabilité par rapport à un rendement esperé r_e s'écrit alors :

$$w^\top \bar{r} \geq r_e \tag{6}$$

La combinaison de ces différentes contraintes et objectifs donne le problème d'optimisation donné dans la publication [1].

3.2.3 Résolution du problème d'optimisation

Ce problème consiste à résoudre un problème d'optimisation sous contraintes de linéarité (pour éviter la solution triviale), et de positivité (pour obtenir des termes w_c positifs).
Il est possible d'utiliser un algorithme basé sur région de confiance (dit "Trust Region"), qui peut être adapté à traiter ce type de contraintes.

3.2.4 Existance de la solution

Il apparait que ce problème d'optimisation possède une solution à condition que le rendement global esperé r_e ne dépasse pas le rendement maximal des actions $r_{max} = \underset{c}{max}(r_c)$. Plûtot que de tester cette condition, il est possible de fixer le rendement espéré sous la forme $r_e = \alpha\, r_{max}$ avec α un paramètre évoluant entre 0 et 1.
Pour $\alpha = 0$, le rendement global du portefeuille sera minimal (juste positif). Toutefois, la diversité pourra être davantage maximisée.
Pour $\alpha = 1$, le rendement global sera maximal. Toutefois, cela se fera au détriment de la diversité : le vecteur solution w_{opt} contiendra uniquement des zeros, sauf pour l'action de rendement maximal.
Ce paramètre peut donc être interprété comme un facteur de risque.

3.3 Apprendre les connexions

La principale contribution de la publication étudiée est un modèle de connexions entre données financières. Cette section explique la démarche des auteurs, ainsi que les points communs entre covariances et connexions.

3.3.1 Motivation

Comme expliqué dans l'introduction, la covariance permet bien de capturer le lien entre deux cours, mais contient également un peu de lien des autres cours. Cela peut fausser les estimation d'inter-dépendances, et par conséquent la construction du portefeuille optimal.

3.3.2 Matrice de connexion

L'idée est de poser le problème d'optimisation de maximisation de diversité par l'utilisation d'une autre matrice que la matrice de covariance. Cette nouvelle matice est appelée "matrice de connections" et représente de façon plus fine les liens entre les différents cours.
Pour ce faire, la publication [1] présente le modèle suivant pour expliquer le rendement d'un cours c à un instant t donné :

$$\hat{r}_{t,c} = \underbrace{a_c + b_c\, r_{t,\Lambda}}_{d_{t,c}} + \sum_{j \neq c} w_{j,c}(r_{t,j} - d_{t,j}) \tag{7}$$

Cette expression fait apparaitre différents termes :
- a_c est une valeur propre à l'action, ne dépendant pas du temps.
- $b_c r_{t,\Lambda}$ représente la dépendance du rendement de l'action par rapport au rendement global. Plus b_k est important, plus les deux sont liés.
- $\sum_{j \neq c} w_{j,c}(r_{j,t} - d_{j,t})$ représente le lien de l'action avec les autres actions. Le terme $w_{j,c}$ représente le poids de la connexion en l'action j et l'action c. Plus il est fort, plus les actions j et c sont liées : cela est proche du concept de covariance.

3.3.3 Apprentissage de la matrice de connexions

L'objectif est maintenant d'estimer les paramètres de la matrice de connexions. Pour cela, il est possible de résoudre un autre problème d'optimisation mathématique, consistant à minimiser l'écart entre l'ensemble des rendements réels $r_{t,c}$ et des rendements modélisés $\hat{r}_{t,c}$.
Pour cela, une méthode consiste à faire de l'optimisation aux moindres carrés, en minimisant les résidus $(r_{t,c} - \hat{r}_{t,c})^2$. Malgré tout, plusieurs contraintes se posent :
- La matrice de connexions $\mathbf{C}$ doit prendre la forme d'une matrice semi-définie positive afin d'être utilisée comme matrice de covariance. Une façon de l'imposer est de l'écrire $\mathbf{C} = \mathbf{P}^\top \mathbf{P}$ où $\mathbf{P}$ est une matrice de taille identique, mais sans contrainte.
- Eviter le phénomène dit de "overfitting", où le modèle cherche à réaliser des estimations coincidant parfaitement aux données d'apprentissage sans réellement détecter de structure entre les données. Pour cela, un terme de régularisation sur les valeurs de a, $\mathbf{P}$ sera ajouté.
- La matrice de connexions doit pouvoir être facilement mise à jour suite à l'ajout de données futurs (t+1, ...). La publication propose d'entraîner le modèle sur les données du premier jour, puis d'ajuster la solution chaque jour.

Dans le cadre de ce projet, l'apprentissage des connexions se fait en deux phases :
- 1) En considérant dans un premier temps uniquement deux dates.
- 2) Puis on part de la précédente solution, et on considère l'ensemble des données.

Cette méthode permet d'avoir rapidement une première solution approximative sur un sous-problème plus simple, puis de considérer l'ensemble du problème avec un assez bon point de départ. Ainsi, on limite le risque de voir l'algorithme converger vers une solution sous-optimale du problème complet.

3.3.4 Ecriture du problème d'optimisation en P

La modélisation des rendements peut se réécrire avec $\mathbf{P}$ sous la forme :

$$\hat{r}_{t,c} = \underbrace{a_c + \sum_v P_{k,v} P_{\Lambda,v}\ r_{t,\Lambda}}_{d_{t,c}} + \sum_{j \neq c} \sum_v P_{c,v} P_{j,v} (r_{t,j} - d_{t,j}) \tag{8}$$

En prenant en compte la pénalisation des termes $a = [a_1, ..., a_C]^\top$, $b = [b_1, ..., b_C]^\top$ et $w = [w_1, ..., w_C]^\top$, la fonction objectif à minimiser s'écrit alors :

$$g(aP) = \sum_{t=1}^{T} \sum_{c=1}^{C} f(r_{t,c})(r_{t,c} - \hat{r}_{t,c})^2 + \lambda(||a||_2^2 + ||\mathbf{P}||_{fro}^2) \tag{9}$$

où $||\mathbf{P}||_{fro}^2$ est la somme des termes de $\mathbf{P}$ au carré, permettant de pénaliser les valeurs de b et w.

λ permet de pondérer cette contrainte : plus λ est élevé, plus l'accent sera mis sur la minimisation des valeurs de a et $\mathbf{P}$, au détriment de la précision du modèle. Si λ est trop faible, la contrainte ne sera pas suffisamment prise en compte, ce qui se traduira par de l'overfitting : le modèle aura tendance à trop coïncider avec les données.

4 Implémentation

Les traitements détaillés dans ce projet sont implémentés en Python pour différentes raison :

- Python étant un langage de haut niveau et interprété, les codes sont par conséquent facilement lisibles et compréhensibles ;
- Python est polyvalent et dispose d'un grand nombre de librairies. Ce projet repose notamment sur NumPy (calcul matriciel), SciPy (librairie scientifique) et MatPlotLib (visualisation) ;
- Ces librairies calquant certaines des fonctionnalités MATLAB, les codes peuvent être facilement portés en MATLAB ;
- Les codes peuvent ensuite être ajustés pour devenir des scripts démarrables de façon automatisée via des scripts PowerShell (Windows) ou Bash (Linux). Ce type de code peut être alors être lancé périodiquement (e.g. une fois par jour), et ses sorties peuvent être redirigées afin d'alimenter d'autres codes (visualisation, aide à la décision...).

4.1 Chargement des données

4.1.1 Données utilisées

Tout d'abord, il est nécessaire de charger les données des différents cours du CAC40 en vue de les exploiter. Des cours ont préalablement été récupérés sous forme de fichiers CSV. Nous utiliserons uniquement dans le cadre de ce projet les prix de clôture de chaque action sur une durée de 10 ans. Il en résulte 40 cours ($1 \leq c \leq C = 40$) en plus de celui de l'indice CAC40 (cours numéro Λ).

4.1.2 Pré-traitements

Un écueil à éviter est de combiner des fichiers de valeurs de cours sans faire attention aux dates. Afin d'éviter cela simplement, il est possible d'utiliser la librairie Python Panda qui permet de charger les fichiers CSV sous forme d'objets dataFrame (tableau de valeurs avec entêtes). Chaque objet dataFrame peut ensuite être combiné aux autres par date pour former un dataFrame global contenant l'ensemble des cours au différentes dates.
Chaque rendement est calculé sur le prix de clôture du cours c à l'instant t $p_{t,c}$ par :

$$r_{t,c} = \frac{p_{t,c} - p_{t-1,c}}{p_{t,c}} \tag{10}$$

4.2 Covariances

La matrice de covariance se calcule simplement via une méthode issue de la librairie Python NumPy.

4.3 Optimisation du portefeuille

Comme motivé plus haut, ce problème d'optimisation est résolu via algorithme de région de confiance, disponible dans la librairie SciPy.
Aussi, l'objectif de rentabilité est implémenté non pas sur valeur, mais sur pourcentage de rendement maximal α. Le paramètre peut être fixé à 0.5 afin d'avoir un bon compromis entre diversité et rentabilité.

4.4 Connexions

4.4.1 Dimensions du problème d'apprentissage

Grâce à l'écriture de la matrice de connexions sous la forme du produit $\mathbf{P}^\top\mathbf{P}$ non contraint sur $\mathbf{P}$, le problème d'optimisation ne pose plus explicitement de contrainte sur la matrice de connexion $\mathbf{C}$. Les paramètres à estimer sont a (taille C) et $\mathbf{P}$ (taille $(C+1)\times(C+1)$). Ces deux paramètres peuvent se combiner en un unique vecteur de paramètres aP (taille C^2+3C+1). Le nombre de paramètre augmentant quadratiquement en fonction du nombre de cours C, il est important de choisir un algorithme d'optimisation adapté aux problèmes de grande taille.

4.4.2 Algorithme d'optimisation

La fonction objectif étant dérivable par rapport à l'ensemble des paramètres contenus dans aP, il est possible d'utiliser un algorithme de type descente de gradient. De par ses performances, l'algorithme BFGS est couramment utilisé pour résoudre ce type de problème. Un des avantages de cet algorithme est qu'il ajuste le pas de descente à la fonction objectif à chaque itération, ce qui évite de fixer le paramètre de taille de pas η présent dans la publication [1].

4.4.3 Initialisation

Comme expliqué dans l'analyse du modèle de rendement, a peut être identifié au vecteur des rendements moyens. On fixe donc $a_0 = \hat{r}$. La matrice de connexions étant basé sur la matrice de covariance, il est possible de fixer $\mathbf{P}$ au départ via une décomposition de Cholesky de la matrice de covariance $\mathbf{Q}$: $\mathbf{P}_0 = Cholesky(\mathbf{Q})$. La matrice de connexions initiale est alors égale à la matrice de covariances : $\mathbf{C}_0 = \mathbf{P}_0^\top\mathbf{P}_0 = \mathbf{Q}$.

4.4.4 Pondération

Les résidus peuvent être pondérés par une fonction, notée $f(.)$ dans la publication, afin d'améliorer la précision du modèle sur une certaine gamme de valeurs de rendement. La figure suivante illustre la différence apportée par la pondération unitaire ($f(r) = 1$) et la pondération dite "top-k" à -10% d'expression $f(r) = \exp\left(-(r+0.1)^2\right)$. Comme illustré dans la figure ci-dessous, cette fonction gaussienne permet d'augmenter le poids des valeurs de rendement avoisinant les -10%, valeur à partir de laquelle on peut considérer que le cours perd suffisamment de valeur. Grâce à cette fonction de pondération, le modèle aura tendance à mieux détecter ces valeurs.

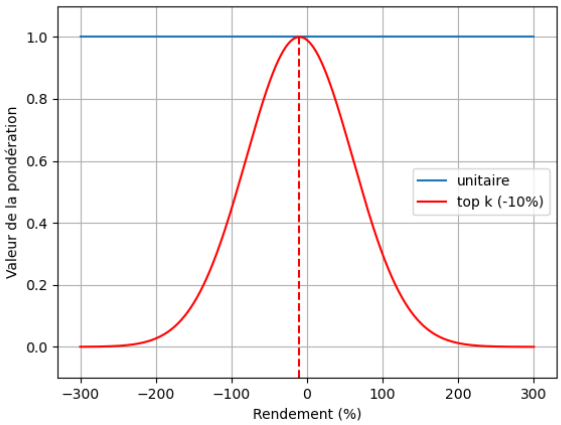

FIGURE 2 – Illustration de la fonction de pondération "top-k" par rapport à la pondération unitaire.

5 Analyse sur données réelles

Le dossier du projet contient trois fichiers Python :

- script_analyse_cours.py : script à paramétrer et à lancer pour résoudre le problème d'apprentissage des connexions et de la minimisation de diversité pour les matrices de covariances et de connexions. Il récupert les données dans un dossier "cours_CAC_40" dans le dossier d'exécution du script, et sauvegarde un fichier "parametres_initiaux.npz".
- fonctions.py : fichier contenant le code des différentes fonctions/algorithme utilisées par le script principal.
- analyse_resultats.py : ouvre le fichier "parametres_initiaux.npz" et en affiche le contenu.

Le script principal a été paramétré avec :

- paramètre de risque $\alpha = 0.5$
- paramètre de régularisation $\lambda = 10^{-4}$
- paramètre d'historique : utilisation des 100 dernières valeurs de cours

5.1 Apprentissage des connexions

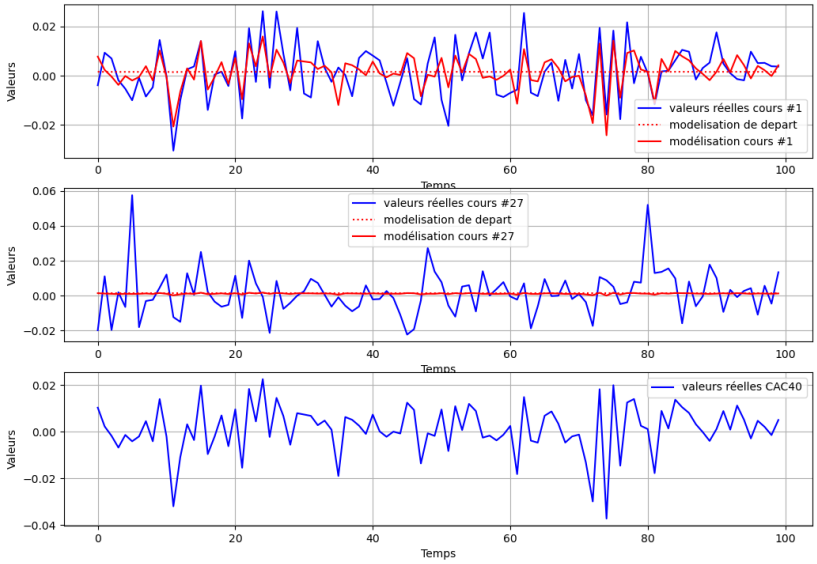

FIGURE 3 – Illustration de la modélisation des cours numéro 1 et numéro 27 en rouge et des données réelles en bleu.

On remarque sur le cours 1 (en haut) que la modélisation au départ (en rouge pointillé, calculée via $\mathbf{P_0}$) est très grossière et n'identifie que la valeur moyenne. Après apprentissage (en trait plein rouge), on constate que le cours est modélisé plus fidèlement par la prise en compte du cours global (en bas) et des valeurs des autres cours (connexions).
Sur le cours 27 (au milieu), on constate en revanche qu'il ne suit pas la tendance moyenne (en bas). Son modèle n'est par conséquent que constitué de termes de connexions aux autres cours. Ces termes étant faibles, la modélisation de ce cours résulte en une valeurs quasiment constante, centrée sur la valeur moyenne (les traits rouge plein et pointillé coïncident).
Il apparaît donc que le cours 27 est particulier, et ne suit pas la tendance globale : cela incite à considérer ce cours pour de la diversification.

5.2 Matrices de covariances et de connexions

Pour rappel, la matrice de connexions (à droite) est initialisée avec la matrice de covariances (à gauche). Le principal point remarquable sur ces matrices est la présence de lignes et colonnes de faibles valeurs. les termes d'une ligne/colonne N sont les termes de covariances et connexions entre un cours N et tous les autres. Si ces termes sont faibles, cela signifie que le cours N est décorrelé des autres.
On retrouve d'ailleurs que le cours $N = 27$ est décorrelé des autres.

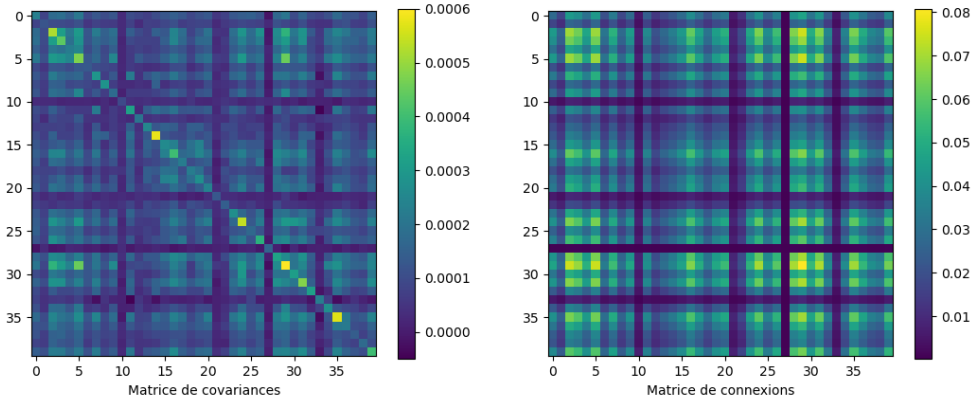

FIGURE 4 – Affichage des valeurs de la matrice de covariances (à gauche) et de connexions (à droite).

Ces matrices sont utilisées pour résoudre le problème de diversification (cf. équation (3)). La figure ci-dessous montre graphiquement le profil de la fonction à minimiser dans le cas où w est de dimension 2. La contrainte sur les poids est donc $w1 + w2 = 1$ et $w1, w2 > 0$.

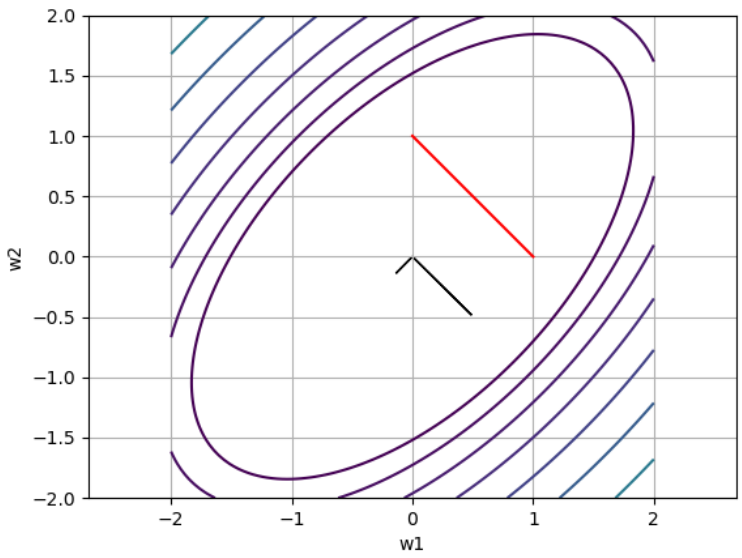

FIGURE 5 – Affichage des lignes de niveau de la fonction de diversification (mauve), de la contrainte sur les poids (rouge) et des vecteurs propres pondérés par les valeurs propres (noir).

Le minimum de la fonction se trouve au centre (0,0), mais n'est pas atteignable au vu de la contrainte (rouge). On remarque que les lignes de niveaux ne sont pas circulaires : cela signifie que la fonction est plus convexe dans certaines directions. Concrètement, cela veut dire qu'investir une certaine quantité sur certains cours aura plus d'impact sur la diversité qu'un investissement similaire sur un autre cours : cela dépend de la convexité de la fonction dans chaque direction.
Cette convexité est donnée par les valeurs propres. La figure ci-dessous donne les valeurs propres des deux matrices classées par ordre décroissant.
Cette figure montre que les valeurs propres de la matrice de connexion s'étalent davantage : le conditionnement (rapport entre valeurs propres maximale et minimale) est nettement supérieur. Ainsi, la représentation des liens sous forme de connexions met clairement en évidence les cours à privilégier (dégradant peu la diversité) de ceux à éviter (dégradant vite la diversité).

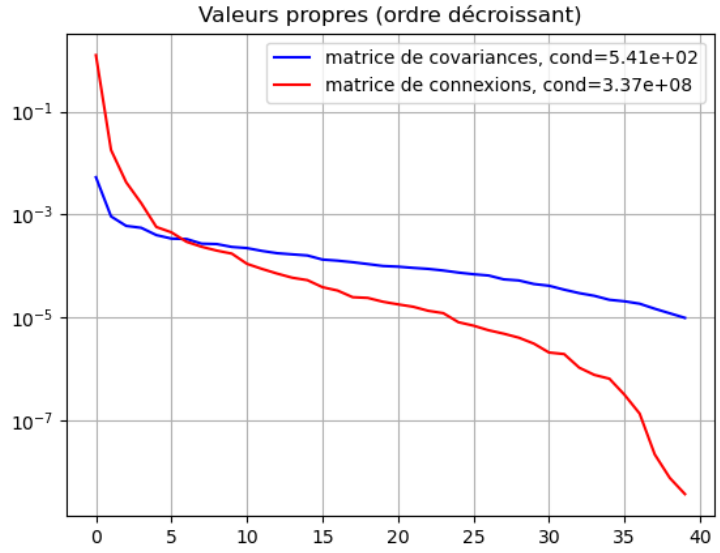

FIGURE 6 – Affichage des valeurs propres des matrices de covariances (bleu) et de connexions (rouge).

5.3 Proportions d'investissement

Pour finir, on regarde les résultats des vecteurs de proportions d'investissement w_Q et w_C donnant respectivement les proportions d'investissement recommandée pour maximiser la diversité et atteindre le rendement espéré r_e. Ces vecteurs sont les solutions du problème de diversification résolu en utilisant la matrice de covariances et celle de connexions, respectivement.

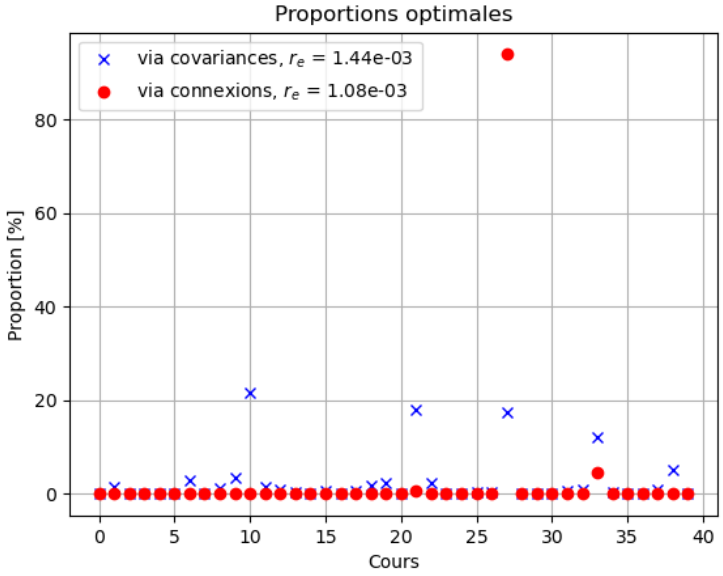

FIGURE 7 – Affichage des proportions d'investissement en utilisant les matrices de covariances (bleu) et de connexions (rouge).

On remarque que l'approche par connexions permet d'obtenir de la diversité avec peu de cours judicieusement choisis. Le rendement espéré r_e est légèrement inférieur, mais sa valeur est plus fiable car obtenue sur des rendements moyens calculés de façon décorrélés.
Ici encore, on remarque que le cours $N = 27$ est manifestement intéressant pour faire de la diversification.

6 Références

[1] Ganeshapillai, G., Guttag, J., & Lo, A. (2013, May). Learning connections in financial time series. In International Conference on Machine Learning (pp. 109-117). PMLR.