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

Dylan Fagot

20 avril 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.

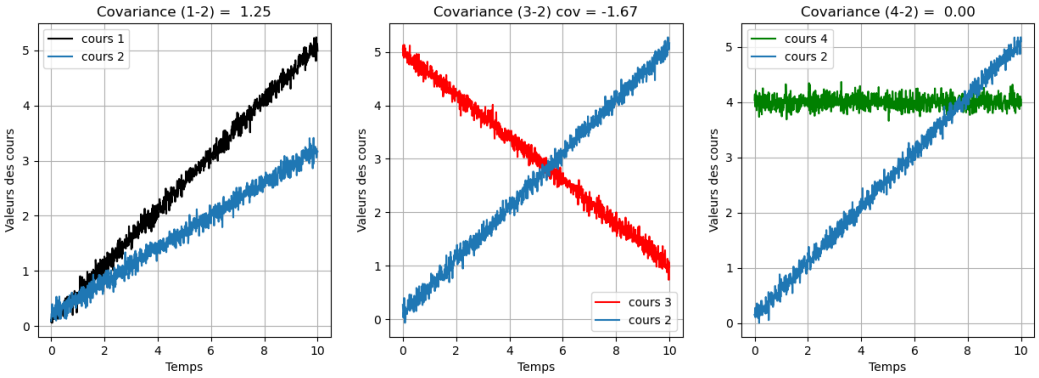

FIGURE 1 – Illustration de trois cas de covariance entre des cours 1 et 2 (gauche), 3 et 2 (centre), 4 et 2 (droite). Le cours 4 (vert) apparait comme décorrelé des cours 1, 2 et 3.

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'entrainer le modèle sur les données du premier jour, puis d'ajuster la solution chaque jour. Une autre stratégie est d'entrainer le modèle sur l'ensemble des données, puis de faire évoluer la solution chaque jour. Celle-ci est plus exigente en terme de temps de calcul au départ, mais donnera une base plus solide de solution lors de la phase dit "online" (prise en compte au fur et à mesure de données d'entrées).

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.

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]. De plus, sa vitesse de convergence est comparable à celle des algorithmes de Newton, ce sans avoir à calculer la matrice Hessienne (taille $C\times C$) à chaque itération.

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 au départ : $\mathbf{P}_0 = Cholesky(\mathbf{Q})$. La matrice de connexion initiale a alors pour valeur $\mathbf{C}_0=\mathbf{P}_0^\top\mathbf{P}_0=\mathbf{Q}$.

4.4.4 Mise à jour

Une fois les paramètres a et $\mathbf{P}$ estimés sur un ensemble de données, il est possible de les utiliser pour initialiser un algorithme de mise à jour des paramètres. L'optimisation de mise à jour reprendra de là et s'ajustera pour prendre en compte les nouvelles données. Cette fonctionnalité n'est pas implémentée à ce jour.

4.4.5 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.

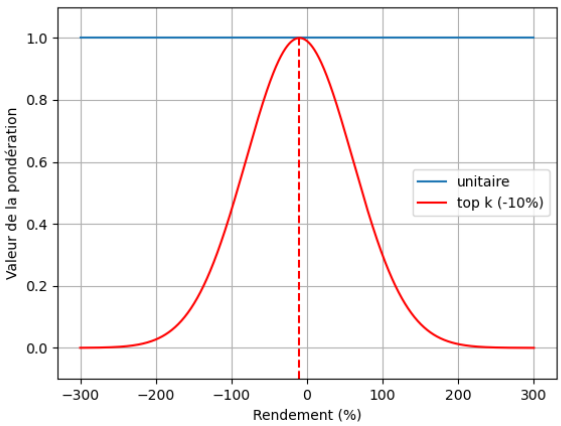

FIGURE 2 – Illustration de la fonction de pondération "top-k" par rapport à la pondération unitaire.

5 Analyse sur données réelles

Visualisation des résultats à venir !

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.