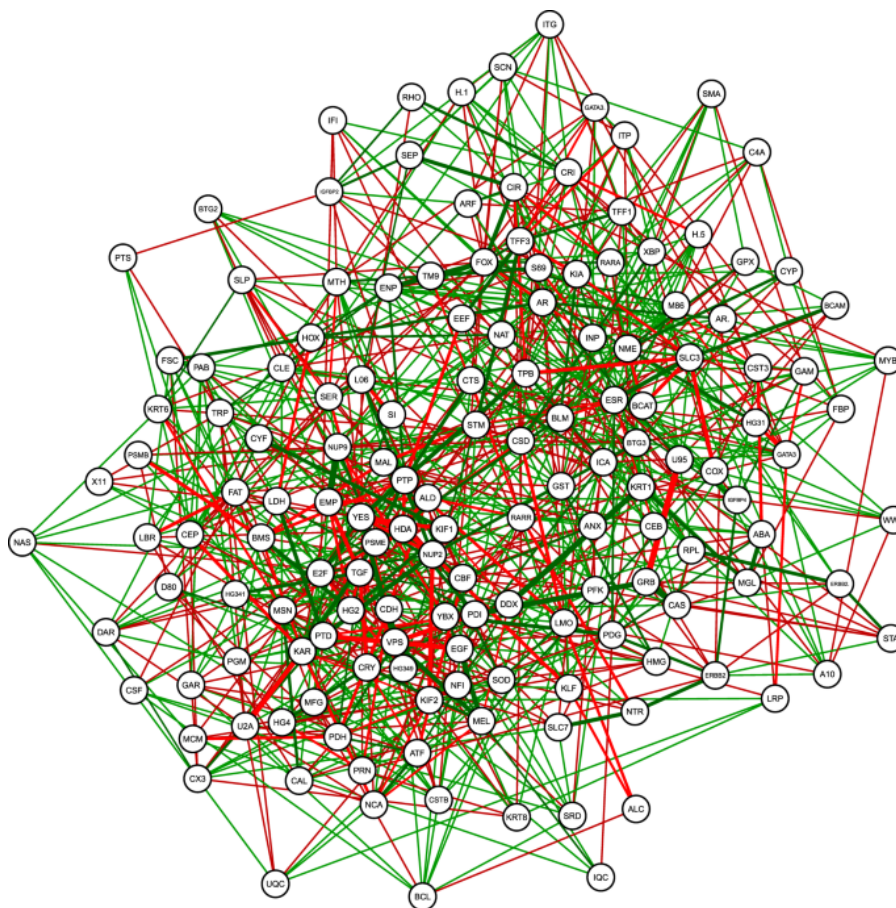

Rapport Pôle Projet

Pôle Data Science - S6

Projet 04 : Modèles Graphiques Parcimonieux pour la Modélisation Financière



Encadré par :
Dr. Maxime FERREIRA DA COSTA

Réalisé par :
Younes-Jihad BOUMOUSSOU
Zakaria BHEDDAR
Aymane Chaoui ADIB
Salma EL ACHKAR
Christophe WANG

Table des matières

1	Introduction	2
2	Généralités sur les graphes gaussiens	3
2.1	Notations	3
2.2	Graphes non orientés	3
2.2.1	Utilité	3
2.2.2	Estimation de la matrice Σ (id. $\Theta = \Sigma^{-1}$)	4
2.2.2.1	Méthode du Maximum de Vraisemblance	4
2.2.2.2	Graphical lasso	5
2.2.2.3	Le rétrécissement Ledoit-Wolf	5
2.3	Comparaison entre les trois estimateurs	6
3	Stationnarité des séries temporelles	7
3.1	Séries temporelles	7
3.2	Stationnarité des processus	7
3.3	Éléments constitutifs d'une série temporelle	8
3.4	Estimation de la tendance par moyenne mobile	8
3.4.1	Estimation empirique de la moyenne mobile :	9
3.4.2	Estimation paramétrique par moindres carrés de la moyenne mobile :	9
3.5	Applications et Simulations	10
3.5.1	Test de stationnarité	10
3.5.2	Stationnarisation d'une série :	11
4	Estimateur de Lasso : Optimisation de l'hyperparamètre λ	12
4.1	Motivation	12
4.2	Méthodes d'entraînement et test	12
4.2.1	La validation croisée - Cross validation	12
4.2.2	La technique du Train-Test Split	13
4.3	Détermination de l'hyperparamètre	13
5	AffinityPropagation	15
5.1	Définition	15
5.2	Algorithme	15
5.3	Implémentation et Points Forts	16
6	Optimisation du portefeuille	17
6.1	Markowitz, la Théorie Moderne du Portefeuille	17
6.1.1	Risque et Rendement Espéré	17
6.1.2	Frontière efficace, définition	18
6.1.3	Frontière efficace, équation	19
6.2	Implémentation	19
6.2.1	Tests sur des vecteurs aléatoirement générés	19
6.2.2	Application sur des actifs réels	20
6.3	Limitations de la Théorie de Markowitz	20
7	Application	22
8	Ouverture et Conclusion	23
8.1	Ouverture sur la prévision des prix d'actifs	23
8.2	Conclusion	25

Abstract

Les modèles graphiques [12] gaussiens sont des outils précieux dans une multitude de domaines, des sciences naturelles à l'économie en passant par les sciences sociales. Ils permettent de modéliser de manière efficace et élégante les relations statistiques entre des variables d'intérêt sous forme de graphes. Ce rapport vise à offrir une introduction pédagogique à ces modèles, mettant en lumière les récents développements dans le domaine de l'estimation du maximum de vraisemblance. Nous commencerons par une exploration des fondements théoriques de cette modélisation, en mettant en avant l'importance de l'optimisation de l'hyperparamètre λ pour une modélisation précise. Ensuite, nous aborderons des aspects avancés tels que l'analyse de la stationnarité d'un processus stochastique, ainsi que des applications concrètes telles que l'optimisation de portefeuille selon Markowitz et la prédiction des prix. Tout au long de ce rapport, nous nous efforcerons d'illustrer comment ces techniques peuvent être mises en œuvre de manière efficace, offrant ainsi des perspectives pratiques pour les professionnels de divers domaines.

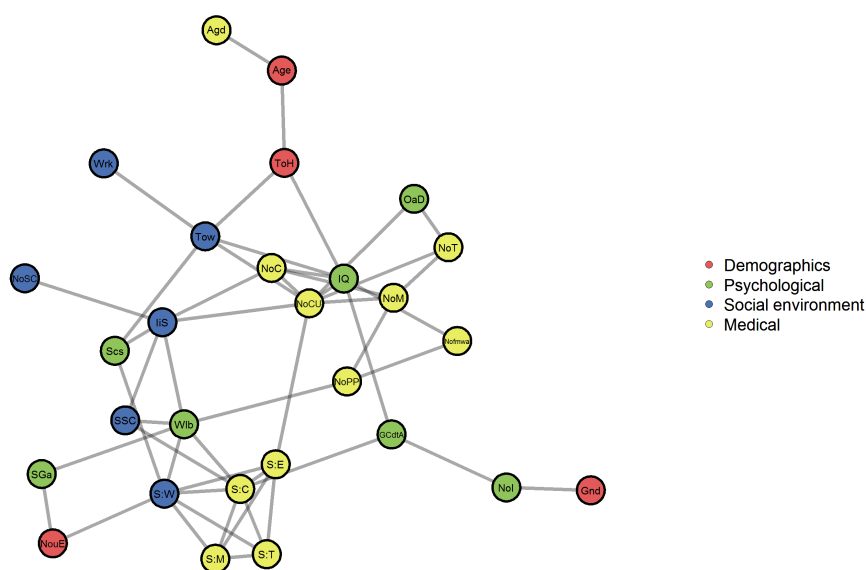


FIGURE 1 – Un exemple de graphe déduit par un modèle graphique gaussien

Chapitre 1

Introduction

Les progrès technologiques de l'ère de l'information ont déclenché une révolution dans la collecte et le stockage de données, permettant une accumulation massive de données à une échelle sans précédent. Cette abondance de données offre une opportunité inestimable pour explorer et comprendre des phénomènes complexes dans une variété de domaines, de la science à l'économie. Dans cette ère de Big Data, l'analyse de ces données nécessite souvent de démêler les relations subtiles entre un grand nombre de variables.

C'est là que les modèles graphiques gaussiens entrent en jeu. En capturant de manière explicite les liens statistiques entre les variables sous forme de graphes, ces modèles fournissent un cadre puissant pour analyser et interpréter des données complexes. Leur utilisation est répandue dans de nombreux domaines, allant des sciences naturelles à la finance, offrant ainsi une méthode universelle pour modéliser des systèmes complexes.

Au cœur de ces modèles se trouve l'hypothèse de la gaussienneté. Cette hypothèse ne se limite pas à une simple commodité mathématique ; elle trouve également sa justification dans des principes fondamentaux de la théorie des probabilités. Par exemple, le théorème central limite stipule que de nombreuses quantités physiques, résultant de la somme de contributions indépendantes, tendent à suivre une distribution gaussienne. De plus, la distribution gaussienne présente une entropie maximale parmi toutes les distributions réelles ayant une moyenne et une covariance spécifiées, ce qui en fait un choix privilégié en termes de parcimonie et de représentation fidèle des données.

En somme, l'utilisation de la gaussienneté dans les modèles graphiques offre à la fois une simplicité mathématique et une représentation robuste des phénomènes naturels. Cette approche s'inscrit dans une perspective plus large où de nombreux systèmes physiques évoluent vers des configurations d'entropie maximale au fil du temps, soulignant ainsi la pertinence et la puissance continue de cette méthode dans l'analyse des données contemporaines.

Chapitre 2

Généralités sur les graphes gaussiens

Dans cette partie, nous expliquerons ce que sont les modèles graphiques gaussiens, comment ils fonctionnent et comment les améliorer. Les modèles graphiques gaussiens sont des outils puissants pour représenter les relations entre les variables sous forme de graphes. Nous verrons comment ils capturent ces relations et comment les ajuster pour obtenir les meilleurs résultats possibles.

2.1 Notations

- Nous désignons par \mathbb{S}^p l'espace vectoriel des matrices symétriques réelles $p \times p$. Cet espace vectoriel est équipé du produit scalaire de trace $\langle A, B \rangle := \text{tr}(AB)$.
- De plus, nous désignons par $\mathbb{S}_{\succeq 0}^p$ le cône convexe des matrices semi-définies positives. Son intérieur est le cône ouvert $\mathbb{S}_{\succ 0}^p$ des matrices définies positives.
- Un vecteur aléatoire $X \in \mathbb{R}^p$ est distribué selon la distribution Gaussienne multivariée $\mathcal{N}(\mu, \Sigma)$ avec les paramètres $\mu \in \mathbb{R}^p$ (la moyenne) et $\Sigma \in \mathbb{S}_{\succ 0}^p$ (la matrice de covariance), si sa fonction de densité [11] est

$$f_{\mu, \Sigma}(x) = (2\pi)^{-p/2} (\det \Sigma)^{-1/2} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}, \quad x \in \mathbb{R}^p.$$

- Nous désignons la matrice de covariance inverse, également connue sous le nom de matrice de précision ou matrice de concentration, par K . En termes de K et en utilisant le produit scalaire de trace sur \mathbb{S}^p , la densité $f_{\mu, \Sigma}$ peut être formulée comme suit :

$$f_{\mu, K}(x) = \exp \left\{ \mu^T K x - \left\langle K, \frac{1}{2} x x^T \right\rangle - \frac{p}{2} \log(2\pi) + \frac{1}{2} \log \det(K) - \frac{1}{2} \mu^T K \mu \right\}.$$

2.2 Graphes non orientés

Un **graphe non orienté** est [1] une paire $G = (V, E)$ où V est l'ensemble des sommets, noté $V(G)$, et E est l'ensemble des arêtes, noté $E(G)$. Chaque arête est une paire non ordonnée $\{u, v\}$ avec $u, v \in V$, indiquant une connexion entre les sommets u et v . Contrairement aux graphes orientés, les arêtes dans un graphe non orienté n'ont pas de direction, ce qui signifie que l'arête $\{u, v\}$ est identique à l'arête $\{v, u\}$.

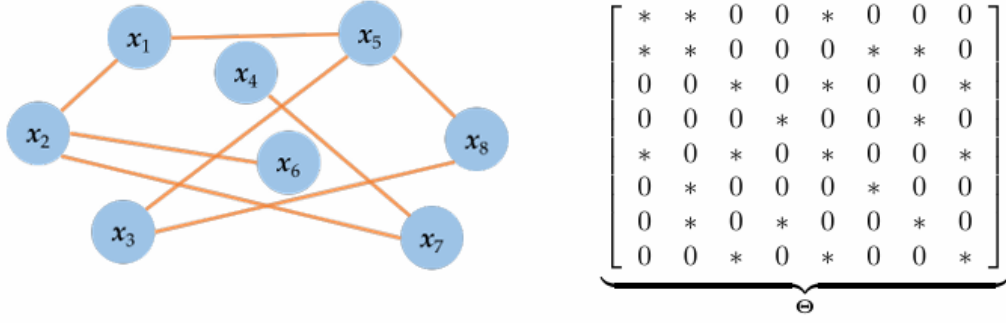
2.2.1 Utilité

La modélisation avec des graphes non orientés, aussi appelés champs de Markov aléatoires, est choisie pour encoder les relations entre les sommets. Dans un graphe de Markov, l'absence d'une arête indique une indépendance conditionnelle entre les variables correspondantes, étant donné les autres variables dans le graphe.

Soit $G = (V, E)$ un graphe non orienté avec les sommets $V = [p]$ et les arêtes E , où $[p] = \{1, \dots, p\}$. Un vecteur aléatoire $X \in \mathbb{R}^p$ satisfait le modèle graphique gaussien (non orienté) avec le graphe G si X suit une distribution gaussienne multivariée $\mathcal{N}(\mu, \Sigma)$ [5] avec

$$(\Sigma^{-1})_{i,j} = 0 \quad \text{pour tout } (i, j) \notin E.$$

Comme dans l'exemple qui suit, pour la matrice de précision $\Theta = \Sigma^{-1}$, le graphe correspondant est illustré dans la figure ci-dessous.



Le graphe G décrit le motif de parcimonie de la matrice de concentration, d'où son nom de graphe de concentration. Les arêtes manquantes dans G correspondent à des relations d'indépendance conditionnelle dans le modèle graphique gaussien associé. Les graphes plus parcimonieux correspondent à des modèles plus simples avec moins de paramètres et plus de relations d'indépendance conditionnelle.

Une autre information est que ce graphe peut être associé à une fonction de coût de corrélation, permettant de définir des arêtes avec des poids donnés pour attribuer de l'importance à celles ayant une forte corrélation. Par conséquent, plusieurs algorithmes peuvent être appliqués pour déterminer les nœuds les plus influents. La formule de corrélation conditionnelle utilisée est la suivante. À noter que cette formule s'applique avec les processus gaussiens [5] :

$$\text{côr}(X_a, X_b \mid X_c : c \neq a, b) = \hat{\rho}_{ab} = \frac{-[\hat{\Sigma}^{-1}]_{ab}}{\sqrt{[\hat{\Sigma}^{-1}]_{aa}[\hat{\Sigma}^{-1}]_{bb}}}$$

Le terme c signifie "sachant les autres", c'est-à-dire la corrélation de a et b sachant les autres variables.

Les modèles graphiques gaussiens sont la contrepartie continue des modèles d'Ising, tous deux des familles exponentielles quadratiques. Ils modélisent uniquement les interactions par paires entre les nœuds, ce qui leur confère une grande flexibilité pour capturer toute structure de corrélation par paires.

En résumé, la détermination de cette matrice permet de construire le graphe associé, soulignant ainsi son importance dans la représentation des relations entre les variables.

2.2.2 Estimation de la matrice Σ (id. $\Theta = \Sigma^{-1}$)

Dans cette partie, nous examinerons plusieurs méthodes d'estimation de la matrice Σ et les comparerons entre elles.

2.2.2.1 Méthode du Maximum de Vraisemblance

La méthode du maximum de vraisemblance (ML) est une approche principale pour estimer les paramètres d'un modèle. Elle vise à déterminer les paramètres qui rendent l'ensemble des observations d'échantillon le plus probable. Bien que l'approche ML soit théoriquement efficace pour un grand nombre d'échantillons, elle peut être moins fiable lorsque les échantillons sont limités, ce qui est souvent le cas en pratique. Malgré ses limites, l'approche ML reste fondamentale et constitue la base des méthodes plus avancées discutées ultérieurement.

Supposons que $\{y_1, \dots, y_n\}$ soient n observations de variables aléatoires continues indépendantes et identiquement distribuées $Y = \{Y_1, \dots, Y_n\}$, chacune avec une fonction de densité $f(y_i; \theta)$, où θ est le paramètre du modèle. La fonction de vraisemblance est la probabilité d'observer les données, définie [3] comme

$$L(\theta; y_1, \dots, y_n) = \prod_{i=1}^n f(y_i; \theta)$$

Comme les observations sont fixes, il est courant d'omettre ces dépendances et d'utiliser la notation $L(\theta)$. La fonction de vraisemblance n'est pas une fonction de densité de probabilité, mais une mesure indiquant dans quelle mesure les données soutiennent un paramètre particulier. Le maximiseur de la fonction de vraisemblance est l'estimation du maximum de vraisemblance (MLE). Nous travaillons avec la fonction de log-vraisemblance négative définie par

$$L(\theta) := -\log L(\theta)$$

Le problème d'estimation des paramètres par maximum de vraisemblance est défini comme suit :

$$L(\mu, \Sigma) = \log \det(\Sigma) + \text{tr}(\Sigma^{-1} \frac{1}{n} \sum_{i=1}^n (y_i - \mu)(y_i - \mu)^\top)$$

En résolvant les conditions du premier ordre pour la moyenne, nous obtenons l'estimation du maximum de vraisemblance $\hat{\mu} = \bar{\mu}$, où $\bar{\mu}$ est la moyenne de l'échantillon. Notons la matrice de précision $\Theta := \Sigma^{-1}$ et la matrice de covariance de l'échantillon.

$$S = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{\mu})(y_i - \hat{\mu})^\top.$$

Remarque : Dans le cas où $n \geq p$ et $S \succ 0$, les conditions d'optimalité du premier ordre de l'équation fournissent des estimations $\hat{\Theta} = \hat{\Sigma}^{-1} = S^{-1}$. Cependant, pour les cas où l'échantillon est de petite taille ($n < p$), on observe que $S \geq 0$, rendant ainsi le problème mal posé. De plus, il est important de souligner que même lorsque $n \geq p$, l'estimation est très probablement mal conditionnée.

En ce qui concerne les avantages de travailler avec la matrice de précision plutôt qu'avec la matrice de covariance, la parcimonie dans la matrice de précision revêt une importance primordiale car elle encode l'indépendance conditionnelle des variables aléatoires. Dans les ensembles de données de grande dimension, la parcimonie dans la matrice de précision est une propriété plausible notée comme étant intrinsèque dans de nombreux domaines d'application. La matrice de covariance est symétrique et semi-définie positive ; cependant, elle doit être définie positive pour que la fonction de densité de probabilité gaussienne existe.

2.2.2.2 Graphical lasso

Le "graphical lasso" est un estimateur de maximum de vraisemblance pénalisé et parcimonieux pour la matrice de concentration ou de précision d'une distribution elliptique multivariée. Il s'agit d'une méthode améliorée et plus efficace que ses prédécesseurs. Initialement développé pour résoudre le problème de sélection de la covariance de Dempster dans le contexte de la distribution gaussienne multivariée, le graphical lasso utilise des algorithmes d'optimisation améliorés pour cette tâche.

L'estimateur de lasso graphique [9] est le $\hat{\Theta}$ tel que :

$$\hat{\Theta} = \arg \min_{\Theta \geq 0} \left(\text{tr}(S\Theta) - \log \det(\Theta) + \lambda \sum_{j \neq k} |\Theta_{jk}| \right)$$

Le paramètre de régularisation λ contrôle le niveau de parcimonie de l'estimateur de lasso graphique $\hat{\Theta}$. En augmentant λ , on obtient une matrice de précision plus parcimonieuse, simplifiant ainsi la structure du modèle graphique en mettant en évidence les relations les plus significatives tout en éliminant les connexions moins importantes.

L'optimisation associée à l'estimateur de lasso graphique est convexe, garantissant un minimum global unique. Cela assure que la solution obtenue est la meilleure estimation possible pour les données fournies et le paramètre λ choisi. De plus, l'algorithme garantit que la matrice de précision estimée $\hat{\Theta}$ est à la fois symétrique et définie positive, assurant la cohérence et la validité statistique du modèle. Ainsi, le lasso graphique fournit un outil puissant et fiable pour l'estimation de la structure de dépendance dans les modèles graphiques.

Remarque : Ce modèle présente certaines limitations que nous aborderons dans les sections suivantes du rapport :

- Le nombre d'échantillons n doit être supérieur au nombre d'actifs ou de sommets.
- Les données sont supposées gaussiennes et stationnaires.
- L'hyperparamètre λ doit être choisi de manière optimale.

2.2.2.3 Le rétrécissement Ledoit-Wolf

Bien que l'estimateur du maximum de vraisemblance soit sans biais asymptotique pour la covariance, il n'est pas fiable pour estimer les valeurs propres de cette matrice. Ainsi, l'inversion de la matrice de covariance pour obtenir la matrice de précision peut être imprécise, voire impossible pour des raisons numériques. Pour remédier à ce problème, une technique appelée "rétrécissement" a été introduite.

Le rétrécissement consiste à ajuster les valeurs propres de la matrice de covariance empirique, réduisant ainsi l'écart entre les plus petites et les plus grandes valeurs propres. Mathématiquement, cela revient à appliquer une transformation convexe qui [8] pénalise.

$$\hat{\Sigma}_{\text{shrinkage}} = (1 - \alpha)\hat{\Sigma} + \alpha \frac{\text{Tr}(\hat{\Sigma})}{p} I_d$$

Choisir le montant de rétrécissement revient à établir un compromis entre biais et variance. Dans leur article de 2004, O. Ledoit et M. Wolf proposent une formule pour calculer le coefficient de rétrécissement optimal qui minimise l'erreur quadratique moyenne entre la matrice de covariance estimée et la vraie matrice de covariance.

2.3 Comparaison entre les trois estimateurs

Pour évaluer les trois estimateurs, nous avons généré une matrice symétrique définie positive avec un coefficient de parcimonie, c'est-à-dire un nombre élevé de zéros (90 %). Cette matrice a été utilisée comme référence pour prédire la même matrice de précision avec ces estimateurs. Pour ce faire, nous avons calculé l'inverse de cette matrice de précision, c'est-à-dire la matrice de covariance. En utilisant la bibliothèque sklearn, nous avons généré des échantillons X gaussiens suivant cette matrice de covariance. Ensuite, nous avons appliqué les trois estimateurs et obtenu les résultats suivants :

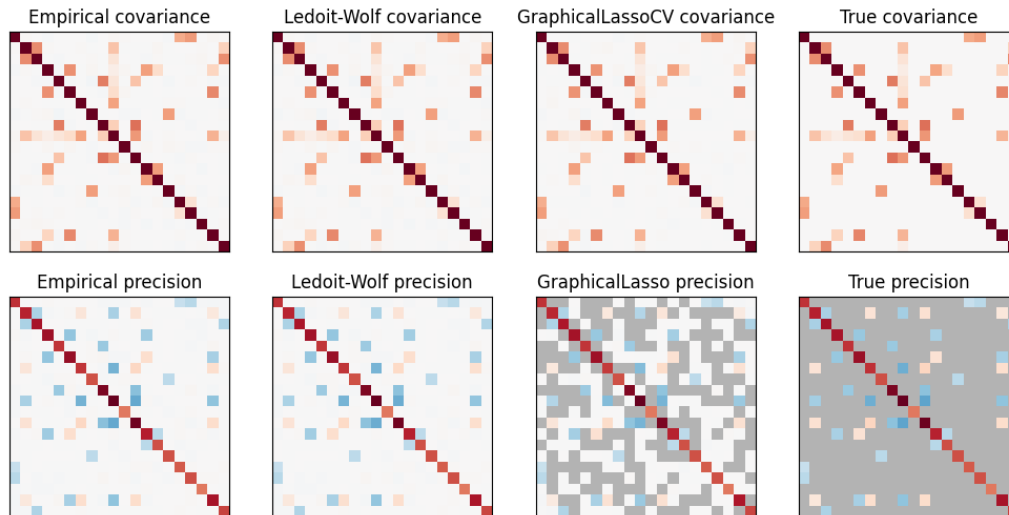


FIGURE 2.1 – Résultats des trois estimateurs

Nous observons que le Graphical Lasso fonctionne bien, comme le montre cet exemple, ce qui renforce notre intérêt à le choisir pour cette étude.

Chapitre 3

Stationnarité des séries temporelles

Dans cette partie, nous modélisons les processus comme étant des processus à temps discret, nous étudierons dans ce cadre des généralités sur la théorie des séries temporelles et des processus stationnaires.

3.1 Séries temporelles

Définition

Une série temporelle (ou série chronologique) est une suite réelle finie $(x_t)_{1 \leq t \leq n}$, $n \in \mathbb{N}$. L'indice t représente une unité de temps (qui peut être le mois, l'année ...)

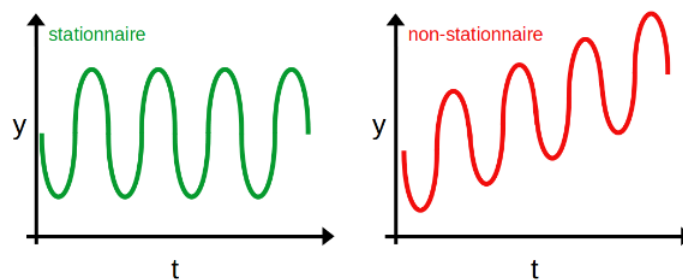
L'objectif de notre étude est de faire des prédictions sur l'évolution des séries à partir des données historiques prises comme entrées.

3.2 Stationnarité des processus

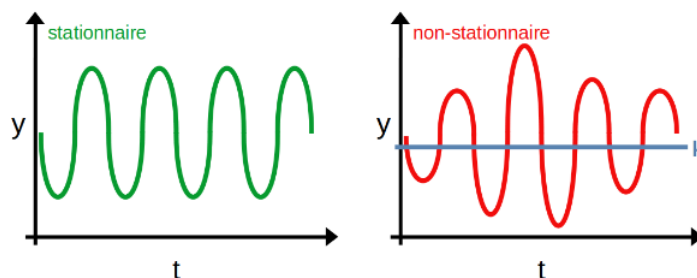
La stationnarité joue un rôle central dans la théorie des processus, car elle remplace (de façon naturelle) l'hypothèse d'observation i.i.d. en statistique.

Qu'est ce que qu'une série stationnaire ?

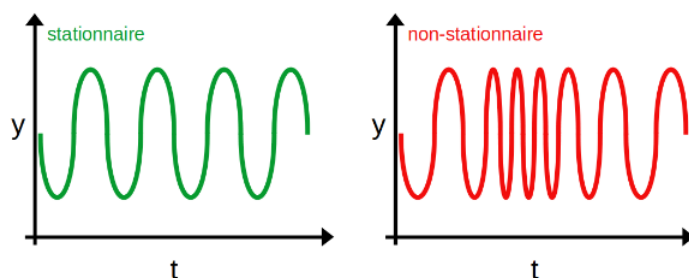
1. La moyenne de la série ne doit pas être fonction du temps.



2. La variance de la série ne doit pas être une fonction du temps.



3. Enfin, la covariance du n^{eme} terme et du $(n+k)^{eme}$ terme ne doit pas être fonction du temps.



Définition

Stationnarité :

Un processus est dit stationnaire au sens faible ou stationnaire au second ordre si :

- La moyenne du processus est constante : $\mathbb{E}(Y_t) = m$ pour tout $t \in \mathbb{Z}$
 - les autocovariances ne dépendent que de la différence entre les observations : $cov(X_t, X_s) = \gamma(|t - s|)$
- Cette dernière propriété implique en particulier que la variance de Y_t est constante : $\mathbb{V}(Y_t) = \sigma^2$

Remarque :

Si une série temporelle est stationnaire et présente un comportement particulier pendant un intervalle de temps donné, on peut supposer sans risque de se tromper qu'elle présentera le même comportement à un moment ultérieur. La plupart des méthodes de modélisation statistique supposent ou exigent que la série temporelle soit stationnaire.

3.3 Eléments constitutifs d'une série temporelle

Une série temporelle est généralement constituée de plusieurs éléments.

- Tendance : représente l'évolution à long terme de la série (échelle interannuelle).
- Saisonnalité : évolution se répétant régulièrement tous les ans.
- Composante stationnaire (ou résiduelle) : ce qui reste lorsque l'on a enlevé les autres composantes. Décrit l'évolution à court terme de la série (échelle journalière).

Le modèle le plus usuel consiste à supposer que la série initiale s'écrit sous la forme (modèle additif)

$$X_t = m_t + S_t + Y_t$$

pour tout $t \in \{1, \dots, n\}$.

avec m_t la tendance, S_t la composante saisonnière (fonction périodique) et Y_t la composante stationnaire.

Objectif

Le but est ici de déterminer les différentes composantes d'une série (X_t) , en particulier, obtenir la tendance de la série qui donne les variations générales de cette dernière.

3.4 Estimation de la tendance par moyenne mobile

La moyenne mobile est une méthode simple permettant d'extraire les composantes basses fréquences d'une série temporelle autrement dit sa tendance.

Elle est également connue comme une méthode de lissage car elle agit comme un filtre passe bas et donc élimine le bruit.

3.4.1 Estimation empirique de la moyenne mobile :

Le calcul de la moyenne mobile dépend d'un paramètre l appelé la largeur de fenêtre. Ce paramètre correspond au nombre d'observations incluses dans le calcul de la moyenne glissante effectuée. Plus l est grand plus le lissage est important (jusqu'à atteindre la fonction constante égale à la moyenne).

La moyenne mobile se calcule ainsi :

$$\hat{m}_t = \frac{1}{2l+1} \sum_{i=t-l}^{t+l} x_i$$

Ceci permet de recueillir la tendance de la série.

3.4.2 Estimation paramétrique par moindres carrés de la moyenne mobile :

Après avoir représenté la série, il est souvent possible d'inférer une représentation paramétrique de sa tendance.

Dans ce cas, on procède par régression (linéaire le plus souvent mais potentiellement non-linéaire) pour estimer cette tendance.

Par exemple, dans le cas d'un processus y admettant une tendance polynomiale d'ordre n :

$$m_t = \sum_{j=0}^n \alpha_j t^j$$
$$y_t = \sum_{j=0}^n \alpha_j t^j + \epsilon_t$$

Seuls les coefficients α_i , pour $i = 1, \dots, n$, sont inconnus et doivent donc être estimés.

L'idée est alors d'estimer les paramètres inconnus α_i , pour $i = 1, \dots, n$ par moindres carrés.

$$\hat{\alpha}^T = (\hat{\alpha}_0, \dots, \hat{\alpha}_n) = \arg \min_{(\alpha_0, \dots, \alpha_n) \in \mathbb{R}^n} \sum_{t=0}^T (y_t - m_t)^2$$

En effet, en introduisant :

$$\mathbb{T} = \begin{pmatrix} m_1 \\ \vdots \\ m_T \end{pmatrix}, \mathbb{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_T \end{pmatrix}, \mathbb{X} = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ t^n & \dots & t^n \end{pmatrix}$$

On peut écrire :

$$\mathbb{Y} = \mathbb{X}\hat{\alpha} + \epsilon \implies \hat{\mathbb{T}} = \mathbb{X}\hat{\alpha}$$

Avec ϵ étant le résidu.

Ainsi un résultat de régression linéaire nous donne que :

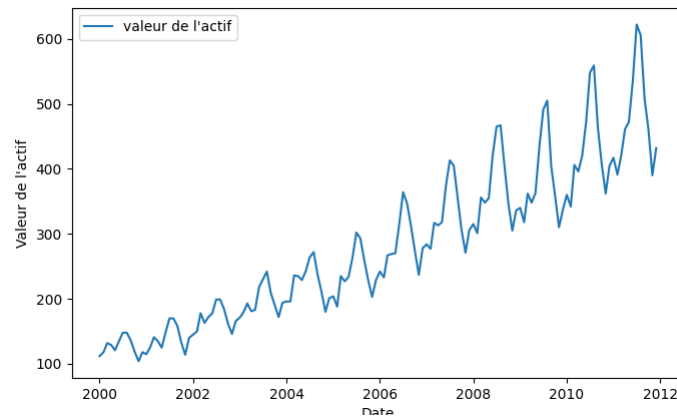
$$\hat{\alpha} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{Y}$$

3.5 Applications et Simulations

3.5.1 Test de stationnarité

Dans cette partie qui sera consacrée principalement à l'implémentation, nous utiliserons la bibliothèque *statsmodels* qui offre un ensemble de fonctions permettant de travailler avec des données de séries temporelles et utilise notamment les résultats théoriques ci-dessus.

Prenons l'exemple de l'évolution de l'actif suivant, avec une échelle arbitraire :

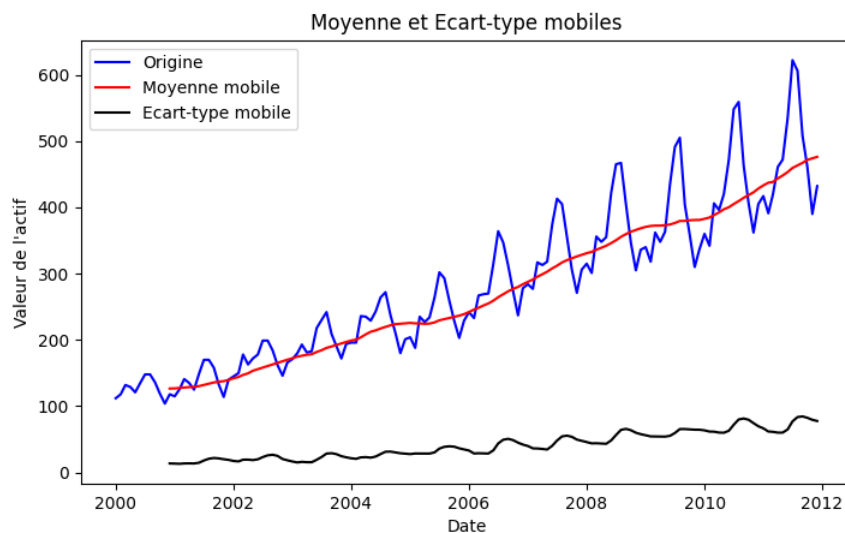


Il y a deux tests principaux pour déterminer si une série temporelle donnée est stationnaire :

- **Statistiques roulantes** : Tracer la moyenne mobile et l'écart-type mobile. Une série temporelle est considérée comme stationnaire si elle demeure constante au fil du temps.
- **Test de Dickey-Fuller augmenté (ADF)** : On considère qu'une série temporelle est stationnaire si la p-valeur est faible et si les valeurs critiques aux niveaux de confiance de 1 %, 5 %, et 10 % sont très proches des statistiques du test ADF (Augmented Dickey-Fuller).

Appliquons ces deux tests à notre série :

1. Premier test :



Résultats :

Comme nous pouvons le voir, la moyenne mobile et l'écart-type mobile varient avec le temps. Nous pouvons donc conclure que la série n'est pas stationnaire.

2. Deuxième test :

```
Statistiques ADF : 0.8153688792060482
p-value : 0.991880243437641
Valeurs Critiques :
1%: -3.4816817173418295
5%: -2.8840418343195267
10%: -2.578770059171598
```

Résultats :

La statistique ADF est loin des valeurs critiques et la p-value est supérieure au seuil (0,05). On peut donc conclure que la série temporelle n'est pas stationnaire.

Conclusion :

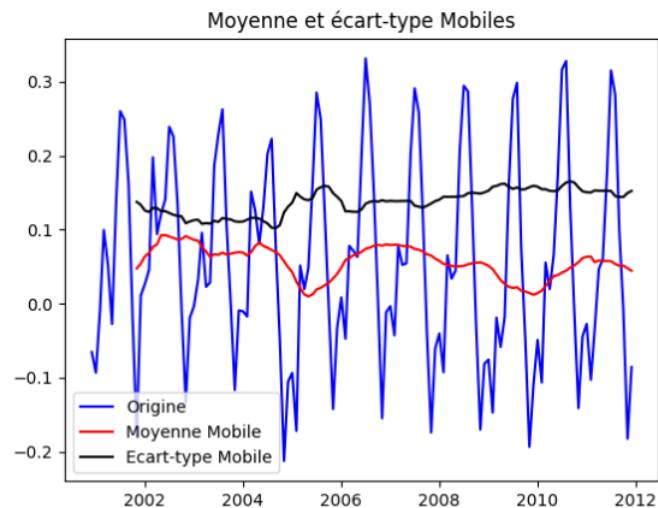
Nous créons une fonction permettant d'exécuter les deux tests qui déterminent si une série temporelle donnée est stationnaire nommée `get_stationarity(timeseries)`.

3.5.2 Stationnarisation d'une série :

Il existe de multiples transformations que nous pouvons appliquer à une série temporelle pour la rendre stationnaire.

Soustraction de la moyenne mobile de la série :

Test 1 :



Test 2 :

```
Statistiques ADF : -3.162907991300869
p-value : 0.022234630001243157
Valeurs Critiques :
1%: -3.4865346059036564
5%: -2.8861509858476264
10%: -2.579896092790057
```

Conclusion :

Comme on peut le voir, après soustraction de la moyenne, la moyenne mobile et l'écart-type sont approximativement horizontaux. La p-value est inférieure au seuil de 0,05 et la statistique ADF est proche des valeurs critiques. Par conséquent, la série temporelle est stationnaire.

Chapitre 4

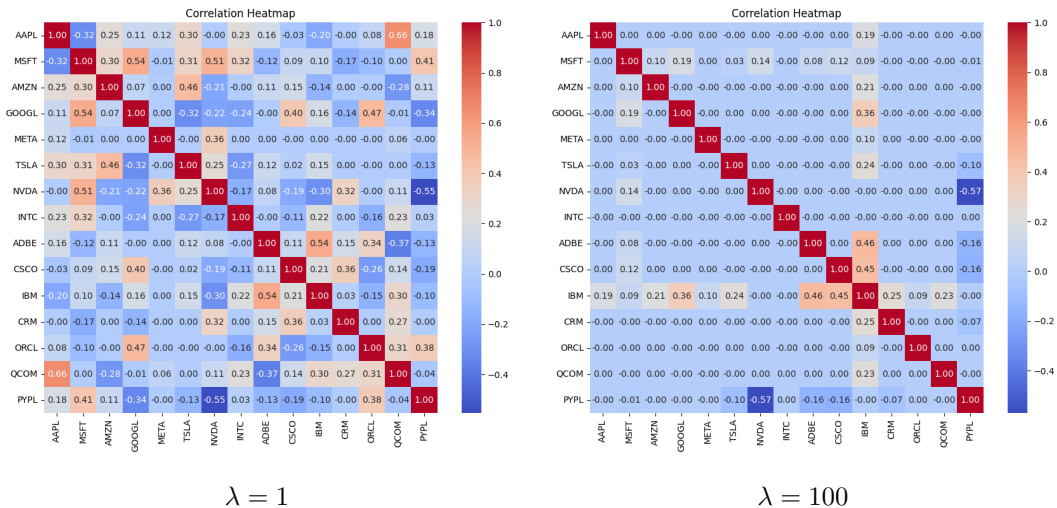
Estimateur de Lasso : Optimisation de l'hyperparamètre λ

Dans cette section, nous nous concentrerons sur le choix de l'hyperparamètre λ utilisé dans le Graphical Lasso. Nous présenterons également la stratégie d'entraînement et le choix du modèle que nous adopterons.

4.1 Motivation

Une des méthodes envisagées consiste à tester un intervalle de valeurs pour λ . En effet, l'expérience montre que λ n'est pas infini et est pratiquement borné. L'idée est donc de définir un score et de comparer les différents modèles obtenus avec différentes valeurs de λ .

Mais pourquoi voulons-nous optimiser ce paramètre ? La raison est simple : ce paramètre contrôle la parcimonie de la matrice. En effet, lorsque λ est petit ou presque nul, le nombre de zéros est très faible, ce qui signifie peu d'indépendances conditionnelles. À l'inverse, si λ est très grand, il y aura beaucoup d'indépendances conditionnelles et de nombreux zéros. Comme le montre cet exemple :



Il est également important de noter que l'optimisation de λ permet d'éviter le surapprentissage. Le surapprentissage, ou overfitting, se produit lorsque le modèle s'ajuste trop bien aux données d'entraînement, capturant le bruit et les fluctuations aléatoires, ce qui réduit sa capacité à généraliser sur de nouvelles données.

4.2 Méthodes d'entraînement et test

4.2.1 La validation croisée - Cross validation

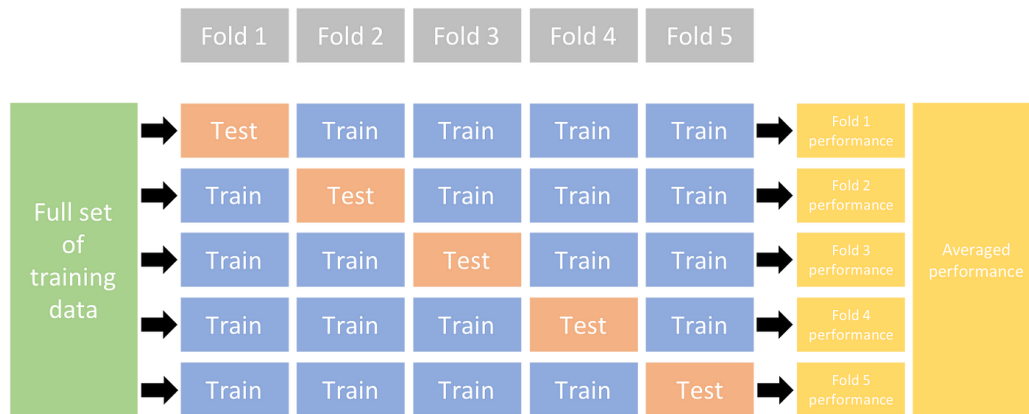
Après avoir entraîné un modèle sur des données étiquetées, il doit être validé pour assurer l'exactitude de ses prédictions en production. La validation consiste à vérifier si les résultats numériques décrivent correctement les relations entre les variables.

Pour évaluer les performances d'un modèle, il est nécessaire de le tester sur des données nouvelles. Selon les résultats, on peut déterminer si le modèle est sous-ajusté, sur-ajusté ou bien généralisé. La validation croisée

[10] (Cross-Validation) est une technique de rééchantillonnage qui permet d'évaluer un modèle même avec des données limitées.

Pour effectuer une validation croisée, une partie des données est mise de côté avant l'entraînement pour être utilisée plus tard pour le test et la validation. Cette méthode est couramment utilisée pour comparer différents modèles et choisir le plus approprié pour un problème spécifique. Elle est simple à comprendre et à implémenter, et elle est moins biaisée que d'autres méthodes.

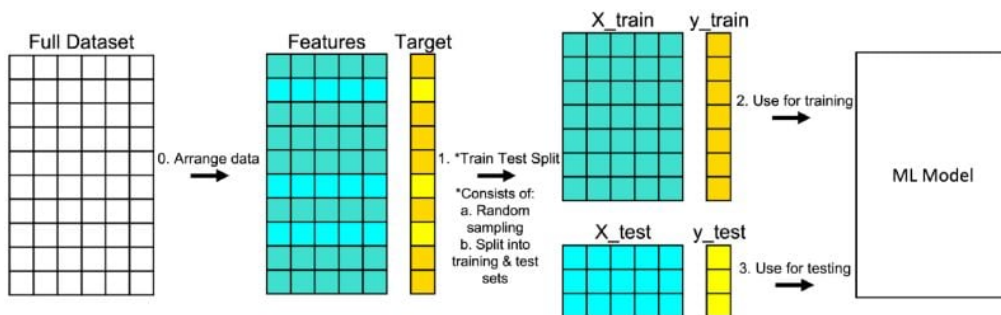
Plus spécifiquement, la méthode de validation croisée appelée aussi méthode K-Folds divise le dataset en K groupes ou "folds". Chaque fold est utilisé une fois comme ensemble de test tandis que les K-1 autres sont utilisés pour l'entraînement. Ce processus est répété jusqu'à ce que chaque fold ait été utilisé pour le test, et la moyenne des scores obtenus est la métrique de performance du modèle. Une valeur K courante se situe entre 5 et 10. Cette technique réduit le biais et est particulièrement utile avec des données limitées.



Ces processus peuvent être effectués manuellement ou à l'aide des fonctions `cross_val_score` et `cross_val_predict` de la bibliothèque Scikit-Learn. La première indique le score de chaque fold, tandis que la seconde prédit les scores pour chaque observation lorsqu'elle faisait partie de l'ensemble de test.

4.2.2 La technique du Train-Test Split

Une autre méthode est la technique Train-Test Split [6] divise aléatoirement le dataset en deux parties : une pour l'entraînement (70-80%) et une pour le test (20-30%). Cette méthode est efficace sauf si les données sont limitées, ce qui peut introduire des biais. Elle est simple à mettre en œuvre et fournit une estimation rapide de la performance du modèle. Cependant, lorsque les données sont limitées, la répartition aléatoire peut entraîner une grande variance dans les performances du modèle, car les échantillons d'entraînement et de test peuvent être très différents les uns des autres. Dans de tels cas, la validation croisée est préférable pour obtenir une évaluation plus robuste du modèle.



4.3 Détermination de l'hyperparamètre

Définition du score

Pour choisir le bon hyperparamètre lambda, nous divisons notre jeu de données en ensembles d'entraînement et de test. Avec un lambda donné, l'ensemble d'entraînement nous permet d'apprendre une matrice de précision. Ensuite, pour évaluer la performance de ce modèle, nous calculons la log-vraisemblance sur l'ensemble de test à l'aide de la matrice de précision estimée. La formule de la log-vraisemblance est :

$$\mathcal{L}(\Theta|X) = \log \det(\Theta) - \text{trace}(S\Theta)$$

Lorsque nous effectuons la validation croisée, le score d'un modèle est calculé comme la moyenne des log-vraisemblances sur les différentes partitions. La formule du score pour la validation croisée est :

$$\text{Score} = \frac{1}{k} \sum_{i=1}^k (\log \det(\Theta_i) - \text{trace}(S_i \Theta_i))$$

Nous cherchons ensuite l'hyperparamètre lambda qui maximise ce score.

Stratégie d'apprentissage revisitée

D'après les éléments précédents, nous pouvons maintenant définir une stratégie d'apprentissage intégrant toutes les informations précédentes. Cette stratégie se déploie comme suit :

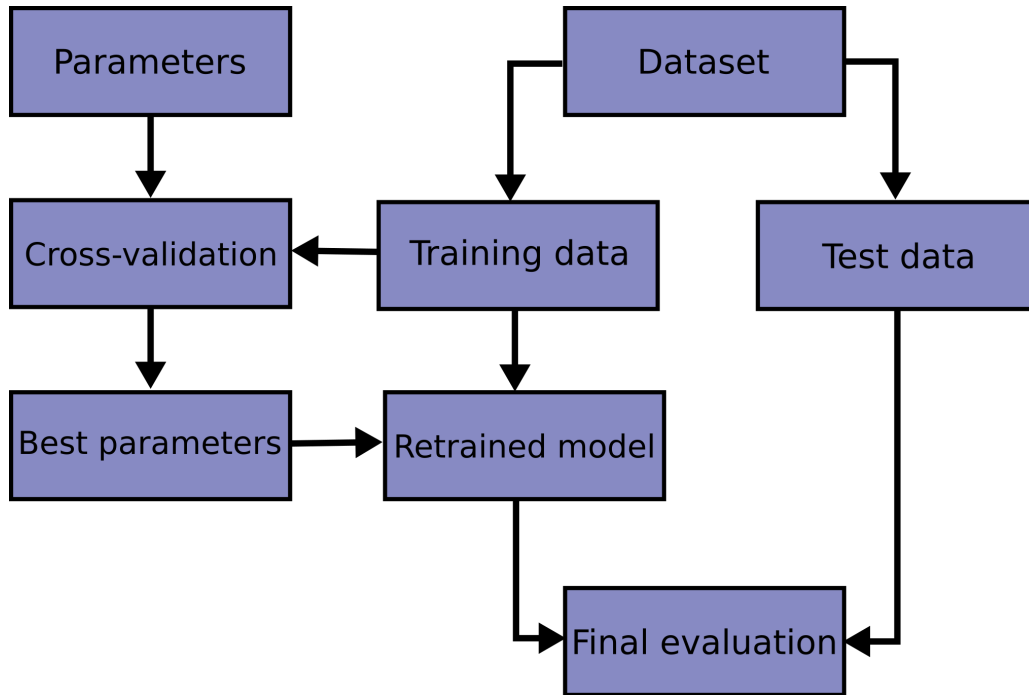


FIGURE 4.1 – Stratégie choisie

Bilan

Faisons un récapitulatif de notre travail jusqu'ici : notre sujet a été d'étudier la théorie des graphes en finance et leur représentation. On a établi une équivalence entre les graphes gaussiens et la représentation sous forme de variables aléatoires gaussiennes. Il se pose là une problème de stationnarité des données qu'on a résolu en chapitre 3. Et dans le cadre d'une approche parcimonieuse à notre problématique, on a utilisé l'estimateur de Lasso pour favoriser la sparsity de notre matrice de données, chose qui a soulevé la problématique d'optimisation d'hyperparamètre et qui a été résolu dans ce chapitre. Ainsi, on est à ce stade parfaitement capable de répondre à la problématique du sujet en produisant un graphe gaussien avec une parcimonie satisfaisante quant aux données . Ainsi, dans les chapitres qui vont suivre, notre groupe a cherché des pistes d'amélioration du sujet actuel, après accord de notre encadrant, et qu'on présente et justifie dans ce qui suit.

Chapitre 5

Affinity Propagation

5.1 Définition

Nous avons appliqué l'affinity propagation pour regrouper les résultats obtenus à partir du graphicalasso. Il s'agit d'un algorithme de clustering qui repose sur le concept de "transmission de messages" entre les points de données. Contrairement à des méthodes comme k-means ou k-medoids, l'affinity propagation ne nécessite pas de pré-déterminer ou d'estimer le nombre de clusters avant son exécution. De plus, tout comme k-medoids, il identifie des "exemplaires", des membres de l'ensemble initial représentatifs des clusters.

5.2 Algorithme

Soit x_1 à x_n un ensemble de points de données, sans hypothèses faites sur leur structure interne, et soit s une fonction qui quantifie la similarité [13] entre deux points quelconques, telle que $s(i, j) > s(i, k)$ si et seulement si x_i est plus similaire à x_j qu'à x_k . Pour cet exemple, la distance au carré négative de deux points de données a été utilisée, c'est-à-dire que pour les points x_i et x_k ,

$$s(i, k) = -\|x_i - x_k\|^2$$

La diagonale de s (c'est-à-dire $s(i, i)$) est particulièrement importante, car elle représente la préférence d'instance, c'est-à-dire à quel point une instance particulière est susceptible de devenir un exemple. Lorsqu'elle est réglée sur la même valeur pour toutes les entrées, elle contrôle le nombre de classes que l'algorithme produit. Une valeur proche de la similarité minimale possible produit moins de classes, tandis qu'une valeur proche ou supérieure à la similarité maximale possible produit de nombreuses classes. Elle est généralement initialisée à la similarité médiane de toutes les paires d'entrées.

L'algorithme procède en alternant entre deux étapes de transmission de messages, qui mettent à jour deux matrices :

- La matrice "responsabilité" R a des valeurs $r(i, k)$ qui quantifient dans quelle mesure x_k est adapté pour servir d'exemple pour x_i , par rapport à d'autres exemples candidats pour x_i .
- La matrice "disponibilité" A contient des valeurs $a(i, k)$ qui représentent à quel point il serait "approprié" pour x_i de choisir x_k comme exemple, en tenant compte de la préférence des autres points pour x_k comme exemple.

Les deux matrices sont initialisées à toutes les zéros, et peuvent être considérées comme des tables de probabilité logarithmique. L'algorithme effectue ensuite les mises à jour suivantes de manière itérative :

1. Premièrement, les mises à jour de responsabilité sont envoyées autour :

$$r(i, k) \leftarrow s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\}$$

2. Ensuite, la disponibilité est mise à jour selon :

$$a(i, k) \leftarrow \min \left(0, r(k, k) + \sum_{i' \notin \{i, k\}} \max(0, r(i', k)) \right) \quad \text{pour } i \neq k$$

et

$$a(k, k) \leftarrow \sum_{i' \neq k} \max(0, r(i', k)).$$

Les itérations sont effectuées jusqu'à ce que les limites des clusters restent inchangées sur un certain nombre d'itérations, ou qu'un certain nombre prédéterminé (d'itérations) soit atteint. Les exemples sont extraits des matrices finales comme ceux dont la "responsabilité + disponibilité" pour eux-mêmes est positive (c'est-à-dire $(r(i, i) + a(i, i)) > 0$).

5.3 Implémentation et Points Forts

Nous avons choisi d'implémenter l'algorithme en utilisant la corrélation comme poids, car elle s'est révélée être un indicateur de similarité robuste entre les points de données. Cette implémentation a été réalisée avec la bibliothèque scikit-learn.

Parlons maintenant des points forts de l'algorithme. Tout d'abord, sa capacité à déterminer automatiquement le nombre de clusters sans nécessiter de spécification préalable. Cette propriété le rend particulièrement utile dans des contextes où le nombre de clusters est inconnu ou variable. De plus, l'algorithme est robuste face à des données de différentes structures et tailles, ce qui lui confère une grande adaptabilité à une variété de problèmes de clustering. Un autre avantage est sa capacité à trouver des exemplaires représentatifs des clusters, facilitant ainsi l'interprétation des résultats. Enfin, sa flexibilité permet d'adapter les critères de similarité en fonction du domaine d'application, ce qui le rend applicable dans divers contextes.

En combinant ces caractéristiques, notre implémentation de l'algorithme d'affinity propagation offre une solution puissante et polyvalente pour le clustering de données.

Chapitre 6

Optimisation du portefeuille

L'objectif de cette partie sera d'étendre le sujet de notre projet d'une simple étude du marché boursier et de l'évolution des actifs à une application concrète des résultats obtenus dans les parties précédentes dans des cas tangibles. Il a en effet été possible de trouver un graphe reliant tous nos actifs financiers d'une relation de corrélation, ce par le biais d'une matrice de précision qu'on a été capable d'estimer.

L'application qui se présente intuitivement à cette étude du marché est l'optimisation d'un portefeuille ; une connaissance assez complète des dynamiques du marché boursier et des risques prévalants aux différentes allocations de ressources permet des allocations moins incertaines, et surtout plus rentables.

C'est donc bien dans ce cadre que s'inscrit cette partie, où il sera question de présenter une méthode d'exploitation de la matrice de précision afin de définir et de déterminer ce que serait une allocation optimale de notre portefeuille dans les actifs considérés. Une simulation de telles allocations sera présentée à la fin où grâce à notre matrice de précision et de bibliothèques Python adaptées, il sera possible de visualiser l'évolution du portefeuille.

6.1 Markowitz, la Théorie Moderne du Portefeuille

La théorie qui sera considérée dans ce qui suit sera la Théorie Moderne Portefeuille (TMP), théorie [2] qui a été introduite par Harry Markowitz dans "Portfolio Selection" en 1952 et qui lui a valu le Prix Nobel en sciences économiques. Celle-ci vise à maximiser le retour sur investissement pour un risque donné, tenant comme variables les poids d'allocation dans notre univers d'actifs. Cette théorie étend le concept de la diversification en investissement, le concept de répartir les allocations sur des actifs les plus variés (et donc décorréllés) possibles pour minimiser le risque, et c'est bien pour cette raison qu'elle semble comme une application plausible aux travaux présentés dans ce rapport.

6.1.1 Risque et Rendement Espéré

La TMP se base sur le fait que les investisseurs sont sensibles aux risques d'un portefeuille ; deux stratégies apportant un même retour sur investissement mais dont l'une est plus risquée que l'autre aboutira au choix de la stratégie la moins risquée. Le choix d'une stratégie plus risquée ne sera possible que si le retour espéré est plus élevé ; la tolérance au risque étant dépendante de caractéristiques individuelles, il va simplement être considéré qu'un investisseur rationnel ne va pas choisir une stratégie d'un investissement si une autre stratégie moins risquée existe.

Le modèle [7] va ainsi considérer que :

- Le rendement d'un portefeuille est la somme des rendements individuels des actifs mis à l'échelle par les poids de l'allocation,
- Le risque d'un portefeuille sera caractérisé par la volatilité σ_p , qui sera fonction de la corrélation ρ_{ij} entre deux actifs i et j (l'indice p représente le portefeuille).

Il est important de remarquer l'importance de la corrélation entre paires d'actifs dans la théorie actuelle, ce qui légitimise son choix dès cette étape de l'étude.

On réécrit les deux hypothèses du modèle comme suit :

$$\mathbb{E}(R_p) = \sum_i \omega_i \mathbb{E}(R_i)$$

$$\sigma_p^2 = \sum_{i,j} \omega_i \cdot \omega_j \cdot \sigma_i \cdot \sigma_j \cdot \rho_{ij}$$

où :

- ω_i représente le poids de l'actif i dans la stratégie ($\sum_i \omega_i = 1$),
- R_i représente le rendement de l'actif i tandis que R_p représente le rendement du portefeuille,
- σ_i représente l'écart type de l'actif i et σ_p la volatilité du portefeuille,
- ρ_{ij} la corrélation entre deux actifs i et j .

On retrouve dans la deuxième équation une traduction analytique du principe de la diversification ; une minimisation du risque vient par le choix d'actifs les plus décorrélés possibles. Si tous les actifs sur lesquels on investit sont des corrélations nulles deux à deux, le risque va se limiter à une somme des variances pondérées par le poids au carré. Par contre, si les actifs sont tous parfaitement corrélés ($\rho = 1$), on va obtenir un risque égal à la somme au carré des écarts pondérés (qui est strictement supérieur). On peut déjà, à cette étape de l'analyse, pressentir qu'une stratégie optimale va choisir une allocation où les actifs sont les plus décorrélés possibles.

6.1.2 Frontière efficace, définition

Le problème d'optimisation élaboré ci-dessus, à savoir maximiser le rendement à risque constant ou minimiser le risque à rendement constant, peut être représenté graphiquement dans un graphe.

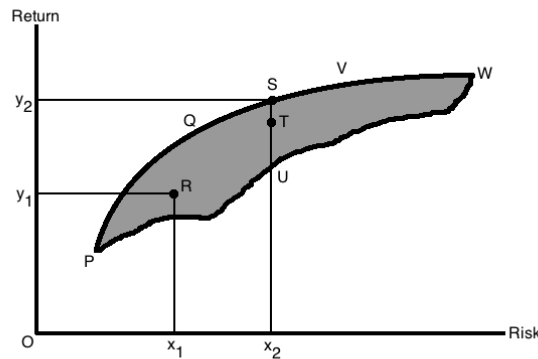


FIGURE 6.1 – Représentation de la frontière efficace sur le plan Rendement vs Risque

La zone en gris comprends toutes les stratégies possibles qu'un investisseur peut choisir. Les portefeuilles efficaces sont ceux qui se situent à la frontière PQVW : c'est ce qu'on appelle frontière efficace. Par exemple, au niveau de risque x_2 , il existe trois portefeuilles S, T, U. Mais le portefeuille S est le portefeuille efficace car il a le rendement le plus élevé, y_2 , par rapport à T et U. Tous les portefeuilles qui se situent à la limite du PQVW sont des portefeuilles efficaces pour un niveau de risque donné.

Tous les portefeuilles situés en dessous de la frontière efficace ne sont pas assez bons car le rendement serait inférieur pour le risque donné. Les portefeuilles situés à droite de la frontière efficace ne seraient pas suffisants, car le risque est plus élevé pour un taux de rendement donné. Tous les portefeuilles situés à la limite du PQVW sont appelés portefeuilles efficaces.

Remarque : Si l'on représente sur le plan Rendement vs Risque l'ensemble des points accessibles, on obtient ce que l'on appelle 'la balle de Markowitz' (Markowitz' bullet). Celle-ci est représenté dans la figure ci-dessous, et c'est un résultat que l'on pourra retrouver.

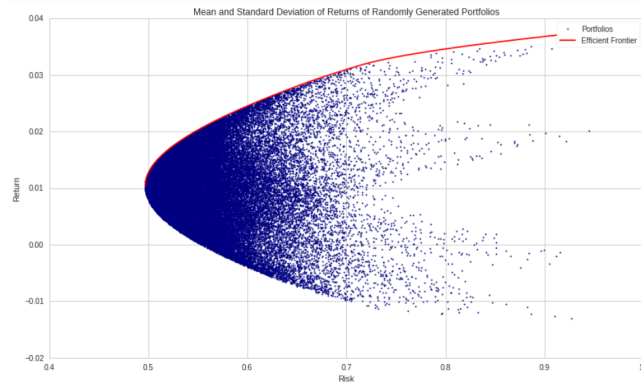


FIGURE 6.2 – Représentation sous Python de la balle de Markowitz

6.1.3 Frontière efficace, équation

Pour une tolérance au risque donnée $q \in [0, +\infty)$, la frontière efficace est trouvée en minimisant l'expression suivante :

$$\omega^\top \Sigma \omega - q R^\top \omega$$

où :

- $\omega \in \mathbb{R}^n$ est le vecteur de poids de la stratégie d'investissement,
- $\Sigma \in \mathbb{R}^{n \times n}$ la matrice de covariance des actifs de l'univers,
- $R \in \mathbb{R}^n$ est le vecteur de rendement espéré.

Ainsi, $\omega^\top \Sigma \omega$ représente la variance du portefeuille et $R^\top \omega$ son rendement espéré. Une telle optimisation trouve le point sur la frontière auquel l'inverse de la pente de la frontière serait q si la variance du rendement du portefeuille au lieu de l'écart type était tracée horizontalement. La frontière dans son intégralité est paramétrique sur q .

Harry Markowitz a développé une procédure spécifique pour résoudre le problème ci-dessus, appelée algorithme de ligne critique, qui peut gérer des contraintes linéaires supplémentaires, des limites supérieures et inférieures des actifs, et qui s'est avérée fonctionner avec une matrice de covariance définie semi-positive. Une méthode alternative serait d'utiliser des multiplicateurs de Lagrange, mais de telles méthodes ne seront pas étayées dans le cadre de ce travail.

Il sera retenu de cette partie son résultat, à savoir l'existence d'un vecteur ω de poids optimal (à risque fixé), et l'existence d'algorithmes permettant le calcul de ce vecteur. Dans ce qui suit, il sera question de présenter les bibliothèques Python permettant l'estimation de ce vecteur. Il reste tout de même important de noter que des langages comme Visual Basic for Applications, JavaScript et d'autres possèdent une implémentation de l'algorithme de ligne critique, tandis que Matlab, Excel, Mathematica ou R, proposent des méthodes génériques d'optimisation.

6.2 Implémentation

6.2.1 Tests sur des vecteurs aléatoirement générés

Une première étude a été réalisée sur un ensemble de n actifs dont les observations ont été générées aléatoirement. Sur cette première approche, il a été généré 50,000 portefeuilles différents, qu'on a tracé sur une courbe Rendement v Risque. Il a été ainsi possible d'observer empiriquement la balle de Markowitz, et l'existence de cette frontière efficace.

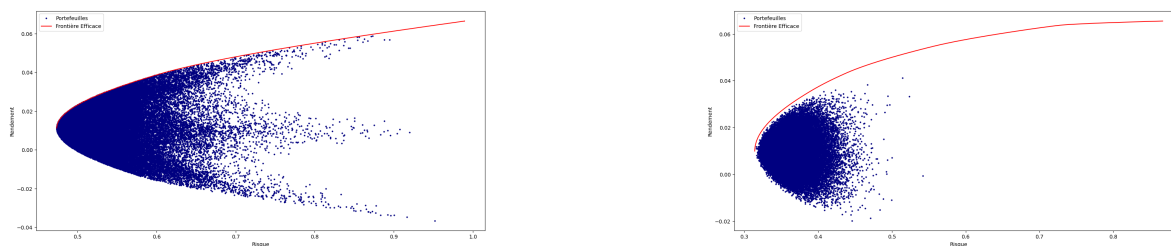


FIGURE 6.3 – Représentation de 50,000 portefeuilles dans un univers à 4 actifs (à gauche) puis à 10 (à droite)

L'équation de la frontière, et donc la solution du problème d'optimisation, a été obtenue grâce à `cvxopt`, une librairie python qui permet l'optimisation convexe.

6.2.2 Application sur des actifs réels

La deuxième application consiste à simuler l'évolution d'un montant de 100k investi en 2010 sur Apple, Microsoft, Google, Nvidia et Amazon. On importe les données grâce à la librairie `yfinance` qui importe des données de Yahoo Finance. On suppose à cette étape qu'on va recalculer des poids pour l'investissement chaque 6 mois et qu'il est suffisant de recueillir les données sur une année. Des périodes justifiées vont être utilisés plus tard dans le cadre de l'application des résultats du chapitre sur la stationnarité des séries temporelles, mais ce qui nous intéresse à cette étape est de visualiser l'évolution d'un portefeuille respectant les critères d'optimisation de Markowitz. On obtiens le résultat suivant

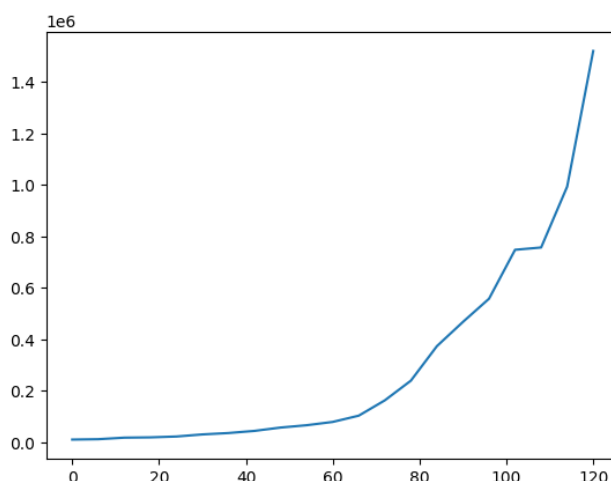


FIGURE 6.4 – Rendement du portefeuille

C'est bien à de tels résultats qu'on s'attends au bout de cette étude.

Remarque : l'absence d'oscillations (ou de bruit) dans notre évolution du portefeuille est due au fait que les dates considérées sont espacées de 6 mois. Ceci masque les petites oscillations observées dans les valorisation boursières et nous donne cette courbe plutôt lisse.

6.3 Limitations de la Théorie de Markowitz

Selon Benoît Mandelbrot, ces théories si simples dans leur application sont complètement déconnectées de la réalité des marchés financiers. Elles ont été souvent critiquées lors, en particulier, des divers krachs boursiers qu'elles ont été incapables de prévoir. Les institutions financières ont adopté des politiques de gestion des risques qui peuvent être considérées comme irresponsables.

Selon Harry Markowitz, le critère moyenne-variance ne propose aucune hypothèse de distribution. Cependant, le principal souci réside dans le fait que les applications qui utilisent ce concept reposent sur la distribution normale, qui sous-estime considérablement les événements « improbables » tels que les crises ou les krachs, alors qu'ils sont en réalité beaucoup plus fréquents que ce que prévoit la théorie. Une autre difficulté majeure réside

dans le fait que les hypothèses sur lesquelles reposent ces théories sont très peu réalistes (notamment en ce qui concerne la rationalité des investisseurs, la continuité et l'indépendance des fluctuations des cours...).

Chapitre 7

Application

Dans cette section, nous travaillons sur plusieurs exemples, qu'ils soient aléatoires ou réels, pour mettre en œuvre la théorie étudiée précédemment. Vous trouverez tous ces fichiers et tout notre travail dans notre repository GitLab à l'adresse suivante : <https://gitlab-research.centralesupelec.fr/christophe.wang/projetfinance.git>.

Chapitre 8

Ouverture et Conclusion

8.1 Ouverture sur la prévision des prix d'actifs

Outre leur rôle dans la modélisation des risques et l'évaluation de portefeuilles, les graphes gaussiens s'avèrent également être un outil précieux pour la prédiction des prix d'actifs financiers. Et c'est dans cette optique que l'axe retenu pour aller plus dans notre sujet a été la prédiction des prix d'actifs. Une recherche complète et appliquée n'a pas été possible pour faire suite aux travaux réalisés dans le cadre de ce projet et les dépassements qu'on a proposé, mais ça reste néanmoins une piste intéressante pour mieux développer notre sujet lors d'une éventuelle suite en S7...

Pour cela, plusieurs méthodes communes ont été envisagées : la première à être présentée est la méthode ARIMA.

Le modèle ARIMA en particulier est un modèle alliant autorégression (AR) et moyennes mobiles (MA) qui a aussi pour avantage de permettre l'étude de séries *non stationnaires*. Elle permet en effet d'étudier des séries présentant des tendances linéaires croissantes ou décroissantes. Ce qui est bien plus réaliste dans le domaine de la finance.

Le principe du modèle AR est le suivant : pour une grandeur Y on suppose que sa valeur à l'instant t est combinaison linéaire de ses n valeurs passées, auxquelles on ajoute un terme d'erreur ϵ_t .

$$Y_t = \sum_{i=1}^n a_i Y_{t-i} + \epsilon_t$$

Le principe du modèle MA est de considérer la valeur à l'instant t comme une combinaison linéaire des erreurs des n valeurs passées.

$$Y_t = \sum_{i=1}^n b_i \epsilon_{t-i} + \epsilon_t$$

Dans le modèle ARIMA [4] on va considérer simultanément ces deux influences.

Néanmoins les modèles AR et MA ne sont valides que sous hypothèse de stationnarité. Le principe du modèle ARIMA est ainsi de ne pas considérer la valeur même de la grandeur Y_t mais une différence afin de remédier à une potentielle instationnarité des données.

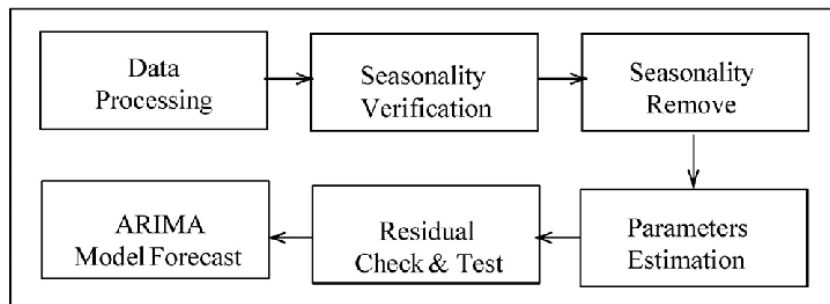


FIGURE 8.1 – Schéma Bloc du fonctionnement de la méthode ARIMA

Il nous reste à déterminer la méthode utilisée pour déterminer les différents coefficients/choisir l'ordre de la différenciation. Ensuite il faudra l'appliquer sur le jeu de données que nous avons obtenu et une fois cela fait, nous pourrions réfléchir à l'implémentation concrète de ces prédictions dans l'optimisation du portefeuille.

Toutefois, d'autres méthodes s'offrent à nous pour étudier dans notre cas la méthode la plus adaptée de prédictions. On trouve ainsi :

- **Les ratios de Fibonacci** : méthode de prédiction du niveau de support et de résistance d'un actif, celle-ci considère le sommet et le creux de la tendance (de 0 à 100) et trace des niveaux de support ou de résistance de la correction théorique au niveaux de Fibonacci (à savoir 23,6 , 38,2 , 50 , 61,8 et 78,6),
- **Les vagues d'Elliott Wave** : Elle précise en particulier que dans chaque tendance à la hausse ou à la baisse, on observe les 5 vagues d'Elliott qui se répètent dans un cycle ininterrompu. Cependant, de la même façon, dans chaque vague, on retrouve toujours les mêmes 5 vagues, et cetera.
- **Les figures chartistes** : cette logique stipule que lorsque les cours évoluent selon une certaine figure, ils devront évoluer ensuite de la même façon que ceux des autres actifs auparavant. À la suite d'un biseau descendant, par exemple, il faudra que les cours évoluent de manière significative en hausse. La compréhension de cette situation et la capacité d'un investisseur à les détecter et à les interpréter peuvent lui permettre de réaliser des bénéfices.

On voit donc bien que le panorama de méthodes de prédiction de prix est assez large, et qu'il rejoint parfaitement notre étude menée comme suite logique à celui-ci. Il s'avère donc comme une piste phare qu'on considère suivre lors de la suite du projet, pour donner un aspect encore plus complet à nos travaux d'études des relations sous-jacentes à nos actifs financiers.

8.2 Conclusion

L'analyse de la problématique de ce projet nous a permis d'apprendre beaucoup de choses en data science et en finance. La modélisation financière ne permet pas seulement de modéliser le comportement des actifs financiers, mais aussi d'obtenir des gains en prévenant et en réduisant les risques. De plus, la data science rend les informations utilisables et capables d'être traitées par des modèles mathématiques. Bien que ces modèles ne soient pas exacts à 100%, chacun essaie de trouver le modèle adéquat qui conduit la plupart du temps à de bons résultats, comme cela a été le cas ici avec l'estimation du paramètre λ et d'autres paramètres.

Ce projet nous a également permis de nous familiariser avec divers outils et techniques de data science, tels que l'analyse exploratoire des données, la visualisation des données, et les méthodes d'apprentissage automatique. En finance, nous avons approfondi notre compréhension des concepts de gestion des risques, de la prévision des prix des actifs, et de l'optimisation des portefeuilles.

Nous avons aussi découvert l'importance de la validation des modèles par la validation croisée pour vérifier la robustesse et la fiabilité de nos prévisions. Ces compétences sont essentielles pour s'assurer que les modèles développés sont non seulement théoriquement solides, mais aussi pratiquement applicables.

Nous tenons à remercier infiniment notre encadrant pour toutes les explications et le soutien qu'il nous a fournis. Son expertise et ses conseils ont été inestimables, surtout lorsqu'il s'agissait de traiter un sujet aussi complexe dès la première fois. Sa capacité à expliquer des concepts difficiles de manière claire et compréhensible a grandement facilité notre apprentissage et notre progression tout au long de ce projet.

Enfin, ce projet nous a également montré l'importance du travail collaboratif et de la communication efficace au sein d'une équipe, deux compétences essentielles dans le domaine professionnel.

Bibliographie

- [1] Quelques rappels sur la théorie des graphes. <https://perso.liris.cnrs.fr/samba-ndojh.ndiaye/fichiers/App_Graphes.pdf/>. [Online; accessed 08-June-2024].
- [2] abcbourse. La théorie moderne du portefeuille. <https://www.abcbourse.com/apprendre/19_theorie_moderne_du_portefeuille.html/>. [Online; accessed 08-June-2024].
- [3] Christophe Chesneau. Sur l'estimateur du maximum de vraisemblance. *Université de Caen*, 2018.
- [4] datascientest. Arima : Modèle de prédiction des séries temporelles. <<https://datascientest.com/arima-series-temporelles>>. [Online; accessed 08-June-2024].
- [5] Christophe Giraud. Introduction to high-dimensional statistics. *Université Paris Saclay*, 2021.
- [6] Marina Kia. Qu'est-ce que la cross validation? <[https://www.jedha.co/formation-ia/cross-validation#:~:text=La%20cross%20validation%20ou%20validation,des%20mod%C3%A8les%20d'apprentissage%20automatique./](https://www.jedha.co/formation-ia/cross-validation#:~:text=La%20cross%20validation%20ou%20validation,des%20mod%C3%A8les%20d'apprentissage%20automatique./>)>. [Online; accessed 08-June-2024].
- [7] mosek. Markowitz portfolio optimization. <<https://docs.mosek.com/portfolio-cookbook/markowitz.html/>>. [Online; accessed 08-June-2024].
- [8] scikit learn. Covariance estimation. <<https://scikit-learn.org/stable/modules/covariance.html#shrunk-covariance/>>. [Online; accessed 08-June-2024].
- [9] Hao Wang. Bayesian graphical lasso models and efficient posterior computation. <<https://projecteuclid.org/journals/bayesian-analysis/volume-7/issue-4/Bayesian-Graphical-Lasso-Models-and-Efficient-Posterior-Computation/10.1214/12-BA729.full/>>. [Online; accessed 08-June-2024].
- [10] wikipedia. Cross-validation (statistics). <[https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))>. [Online; accessed 08-June-2024].
- [11] wikipedia. Loi normale multidimensionnelle. <https://fr.wikipedia.org/wiki/Loi_normale_multidimensionnelle/>. [Online; accessed 08-June-2024].
- [12] wikipedia. Modèle graphique. <https://fr.wikipedia.org/wiki/Mod%C3%A8le_graphique/>. [Online; accessed 08-June-2024].
- [13] Okan Yenigün. The mechanics of affinity propagation clustering. <<https://python.plainenglish.io/the-mechanics-of-affinity-propagation-clustering-eb199cc7a7c2/>>. [Online; accessed 08-June-2024].