RAPPORT DE PSC - ECO03

21 septembre 2023

Artur César ARÁUJO ALVES
Constant LE BEZVOËT
João Pedro SEDEU GODOI
João Pedro TANAKA MONTALVÃO
Rafael Ryoma NAGAI MATSUTANE

SUMMARY

1 INTRODUCTION

Pour comprendre un phénomène et ses causes, nous pouvons utiliser un modèle qui a pour entrées les caractéristiques et les résultats passés de ce phénomène pour essayer de prédire des résultats futurs à partir de données actuelles. Par exemple, nous pourrions créer un modèle pour prédire la probabilité qu'une personne développe une maladie spécifique à partir de ses caractéristiques, telles que l'âge, le poids, les antécédents familiaux et la pratique d'activités physiques. Cependant, toutes les caractéristiques que nous prenons n'influencent pas nécessairement le résultat final. C'est pourquoi, dans un contexte de rareté des données ou de faible puissance de calcul, il est important de détecter comment les caractéristiques influencent le phénomène étudié.

Pour parvenir à cet objectif, il convient d'utiliser ce que l'on nomme les facteurs plutôt que les caractéristiques. L'analyse factorielle est une technique qui vise à identifier les facteurs explicatifs cachés dans un ensemble de variables. Cette technique est très importante en statistique car elle permet de réduire la dimensionalité du problème, en supprimant des redondances dans les effets causés par les variables. En outre, l'identification du nombre de facteurs ou de leur nature peut s'avérer très utile pour mieux expliquer un phénomène complexe.

Dans ce contexte, il est important de différencier la signification des caractéristiques et des facteurs. Les caractéristiques sont les variables observables d'un modèle, tandis que les facteurs sont les variables « cachées » qui causent ou influencent le phénomène étudié.

Notre rapport présente une étude sur une méthode récente appélée *Projected PCA* basé sur cette analyse à facteurs. Cette méthode permet d'estimer les facteurs cachés et leurs coefficients ainsi que de réaliser des prédictions à partir de l'ensemble de caractéristiques. Notre objectif a été de nous familiariser avec cette nouvelle méthode, de mettre en oeuvre notre propre implémentation en y apportant des améliorations, et d'appliquer des outils de validation supplémentaires par rapport à l'article de référence. Nous avons également souhaité appliquer ce modèle à des données réelles dans un contexte plus large, où les caractéristiques ne sont pas nécessairement invariantes dans le temps, une hypothèse qui peut être difficile à justifier en finance.

1.1 Modèles à facteurs dans la finance

L'analyse factorielle est très importante en finance car elle essaie de comprendre comment les caractéristiques visibles sont liées les unes aux autres. Généralement cette analyse est utilisée pour modéliser le rendement ou le risque d'un actif.

L'un des premiers modèles que nous avons trouvés dans la littérature et qui utilise cette idée est le modèle *Capital Asset Pricing Model* (CAPM) [15]. Dans ce modèle, le rendement attendu ER_i d'un actif est modélisé comme une fonction du taux « sans risque » R_f (généralement des obligations d'État) et du rendement β associé au risque de l'actif et du marché sur lequel il se trouve, qui a un rendement attendu ER_m.

$$ER_i = R_f + \beta_i(ER_m - R_f)$$

Bien que très simple, ce modèle est l'un des piliers de la théorie financière moderne et est largement utilisé pour étudier le rendement associé au risque d'un actif dans un portefeuille.

D'autres modèles factoriels ont été créés par la suite, par exemple, Eugene Fama a développé le modèle à trois facteurs [1], qui identifie trois facteurs prédominants dans les rendements des actifs, qui sont :

1. Le risque de marché, déjà identifié précédemment par le modèle CAPM.
2. La meilleure performance relative des entreprises à faible capitalisation boursière par rapport à celles à forte valeur.
3. La meilleure performance relative des entreprises à forte valeur book-to-market par rapport à celles à faible valeur book-to-market. En résumé, les entreprises dites de « valeur », c'est-à-dire celles dont la valeur de marché est faible par rapport à leur valeurs comptables, ont tendance à obtenir de meilleurs résultats que les entreprises de « croissance », dont la valeur de marché est très élevée par rapport à leurs valeurs comptables, ce qui implique que le marché s'attend à ce que ces entreprises se développent.

En raison de ces caractéristiques, le rendement des actifs peut être modélisé comme suit :

$$r = R_f + \beta(R_m - R_f) + b_s \cdot SMB + b_v \cdot HML + \alpha$$

Dans cette équation, les premiers termes sont les mêmes que dans le CAPM, SMB (*Small Minus Bigger*) représente le facteur taille et HML (*High Minus Low*) représente le facteur valeur des entreprises. Ces facteurs nous aident à comprendre les caractéristiques des actifs qui sont réellement liées aux rendements.

Le modèle à trois facteurs mentionné ci-dessus illustre bien le pouvoir explicatif des modèles factoriels tel que mentionné dans [3] : les rendements plus élevés des actions de valeur par rapport aux actions de croissance ne peuvent pas être expliqués à partir d'une analyse fondamentaliste, mais sont parfaitement explicables lorsque nous les analysons en tant que l'un des facteurs du modèle. Par contre, un échec liée à ce type de modèle c'est l'estimation des coefficients utilisés pour décrire le modèle : la plupart d'estimation étant faite en utilisant la régression linéaire, nous observons que l'erreur associé à cette estimation augmente considérablement quand la corrélation entre les coefficients et les variables observables ne sont pas trop linéaire, **comme nous allons montrer dans la suite de notre travail.**

1.2 État de l'art de factor models

Comme est fait pour les modèles à 3 ou 5 facteurs d'Eugène Fama, une généralisation à K facteurs est possible. Si l'on note $\{y_{it}\}_{i\leq p, t\leq T}$ avec p et T respectivement la dimension et la taille de l'échantillon des données, une façon plus génerale de formuler un modèle à K facteurs serait :

$$y_{it} = \sum_{k=1}^{K} \lambda_{ik} f_{tk} + \mu_{it}, \quad i = 1, \ldots, p,\ t = 1, \ldots, T$$

Où les f et λ sont les facteurs non observables et leurs coeficients de charge (*loading factors*) respectifs et les μ_{it} sont les bruits associés à chaque actif.

Avec les progrès des techniques d'apprentissage machine et la grande quantité de données dont nous disposons actuellement, il était nécessaire de créer des outils pour détecter et calculer les facteurs d'une manière plus générale. La découverte de ces facteurs est particulièrement importante dans ce domaine car elle réduit la quantité de données utilisées pour entraîner un algorithme, ce qui le rend plus efficace et évite les redondances. En ce qui concerne cet aspect, l'un des modèles les plus utilisés est basé sur la *principal component analysis*(PCA), qui, en considérant l'espace des données comme un espace vectoriel, cherche à trouver une nouvelle base dont les axes donnent une meilleure explication des données, ce qui permet de diminuer le nombre de facteurs observés et donc la puissance de calcul nécessaire.

Cette technique simple, qui peut être implementé facilement à l'aide d'outils d'algèbre linéaire, peut ensuite être complexifié pour prendre en compte de nouveaux phénomènes ou pallier à ses faiblesses, comme la difficulté d'incorporer des informations de natures différentes au sein du modèle , la nécessité de considérer que les coefficients sont fixes au cours du temps ainsi que le besoin d'avoir beaucoup de données étalées dans le temps pour l'entraîner.

La plupart des techniques existantes travaillant avec le modèle à facteurs, comme le PCA, reposent sur une longue fenêtre temporelle d'échantillonnage, c'est-à-dire un T assez grand pour avoir niveau de précision acceptable. Plus techniquement, les meilleurs taux de convergence pour l'estimation des coefficients trouvés sur la littérature sont de l'ordre de $O_p(T^{-1/2})$ [8]. Cependant, même avec la disponibilité des « *Big Data* », avoir des grands T dans les applications financières n'est pas pratique car nous sommes généralement obligés d'utiliser un échantillonnage mensuel pour réduire les corrélations sérielles, ce qui réduit considérablement le nombre d'échantillons.

Outre la finance, les puces génétiques, l'imagerie médicale et la reconnaissance de texte sont des exemples d'autres domaines qui présentent les mêmes caractéristiques de *high-dimension, low-sample-size* (HDLSS), pour lesquelles le PCA classique n'est pas si efficace[12].

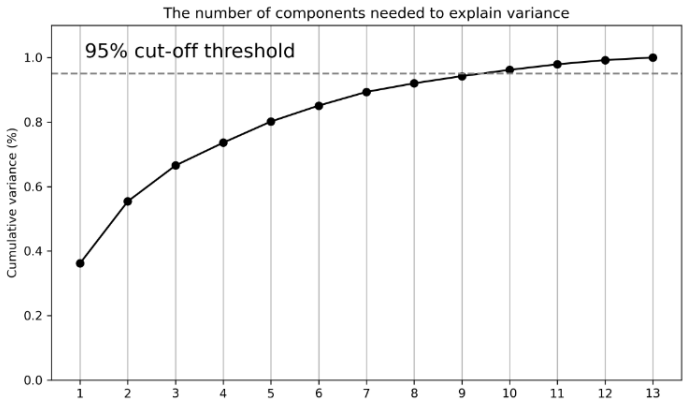


Figure 1 – Pouvoir explicatif des composantes de la PCA dans l'ensemble de données Wine Dataset, qui comporte 13 caractéristiques au total.

2
OBJECTIFS DU TRAVAIL ET MÉTHODOLOGIE

Dans cet esprit, notre travail a deux objectifs : déterminer le nombre K de facteurs cachés et explorer une proposition récente d'algorithme basé sur la PCA classique : la PCA projetée (P-PCA)[8], une méthode capable d'accélérer considérablement les taux de convergence des estimations des coefficients des facteurs. Le grand avantage est que la convergence est cette fois atteinte même si T est petit et en contrepartie l'exigence tombe sur la dimension de l'échantillon, c'est-à-dire une grande valeur pour p. Plus précisement, les taux de convergence pour l'estimation des *loading factors* obtenus avec la P-PCA sont de l'ordre de $Op((pT)^{-1/2})$. Ce compromis est très avantageux dans les applications financières puisqu'il est plus fréquent de trouver des situations du type haute dimension et basse taille d'échantillon.

Pour acquérir des connaissances sur les modèles d'évaluation des actifs, nous nous sommes répartis le travail de telle façon que chacun a été responsable pour un article entre les cinq mentionnés dans la proposition détaillée[1][2][3][4][5] et nous avons presenté un article par semaine. Ces études initiales ont servi à comprendre les premiers aspects de la finance quantitative et introduire les membres du groupe à ce sujet.

Après ces premières études, le groupe s'est concentré sur l'étude des articles relatifs au *Projected-PCA*[8] et à la simulation des données[7]. Étant donné que les articles font réference au modèle choisit pour le projet, tout le groupe a realisé la lecture et des rencontres ont été faites avec le tuteur pour clarifier nos doutes.

Pour cela, nous avons divisé le groupe en deux fronts : la simulation de données et la P-PCA. Même si nous disposons déjà des données réelles nécessaires à l'évaluation du modèle, le premier groupe est important pour que nous puissions comprendre l'influence de la P-PCA

dans un phénomène dont nous connaissons les facteurs et la relation entre eux et le résultat final.

En utilisant la méthode P-PCA et les données simulées, nous réalisons des expériences pour analyser la performance des differents estimateurs de K, bien comme pour analyser la performance dans l'estimation des courbes des fonctions caractéristiques du modèle à facteurs utilisé. Pour ces expériences, nous utilisons des valeurs de K qui sont différents de 3 pour un nombre de caractéristiques plus grand que 1.

Finalement, nous réalisons des expériences pour analyser les différences entre les méthodes P-PCA et PCA classique, bien comme appliquer la méthode P-PCA sur les données réelles pour analyser sa performance en comparaison avec les autres méthodes de prédiction.

3
P-PCA

Le P-PCA (*Projected PCA*) est une méthode appliquée à la théorie des facteurs basée sur la méthode traditionnelle de *PCA* [13]. L'objectif principal est d'estimer les facteurs latents et les coefficients correspondants à partir des caractéristiques et des rendements. La seule différence avec son prédécesseur est que les valeurs de retour Y sont projetées sur un espace fonctionnel approprié généré par les caractéristiques X.

3.1 *PCA* dans les modèles à facteurs

Avant de détailler la technique *P-PCA*, il est intéressant de présenter brièvement l'analyse en composantes principales traditionnelle lorsqu'elle est appliquée au modèle à facteurs examiné ici. Nous considerons la matrice $\mathbf{Y}$ de taille $p \times T$ qui contient les valeurs des rendements attendus pour p actifs en considérant un intervalle de temps T :

$$\mathbf{Y} = \mathbf{\Lambda F}' + \mathbf{U}$$

L'objectif est d'estimer les matrices $\mathbf{F}$ et $\mathbf{\Lambda}$ de taille $T \times K$ et $p \times K$, correspondant respectivement aux facteurs et à leurs coefficients.

Il convient tout d'abord d'observer que, sans autres restrictions, il est impossible d'identifier ces deux matrices individuellement, puisque pour tout $\mathbf{H} \in GL_k(\mathbb{R})$, $\mathbf{\Lambda H^{-1}}$ et $\mathbf{HF}$ sont des solutions si $\mathbf{\Lambda}$ et $\mathbf{F}$ sont aussi des solutions. Pour cette raison, nous utilisons les hypothèses suivantes, également appelées conditions d'identifiabilité, qui correspondent à la PC1 en [13] :

$$\begin{cases} \frac{1}{T}\mathbf{F}'\mathbf{F} = \mathbf{I_k} \\ \mathbf{\Lambda}'\mathbf{\Lambda} \text{ est une matrice diagonale avec des entrées distinctes} \end{cases}$$

Dans ces conditions, nous pouvons finalement identifier les deux matrices à partir de leur produit. Pour comprendre comment cela est réalisable, considérons une simplification dans laquelle l'erreur idiosyncratique $\mathbf{U}$ (c'est à dire, l'erreur qui ne peut pas être expliqué par les caractéristiques) est nulle. Dans ce cas, nous pouvons regarder le produit de Y avec sa transposée :

$$\mathbf{Y}'\mathbf{Y} = \mathbf{F\Lambda}'\mathbf{\Lambda F}'$$

Compte tenu des hypothèses faites, l'expression ci-dessus est très proche d'une décomposition en vecteurs propres, ce qui nous conduit naturellement à identifier $\mathbf{F}$ à partir de l'analyse en composantes principales. En d'autres termes, nous considérerons $\mathbf{F}$ à une constante près,

comme étant une matrice dont les colonnes sont les K vecteurs propres du produit $\mathbf{Y}'\mathbf{Y}$ correspondant aux K plus grandes valeurs propres.

L'amplitude de la constante peut être fixée à l'aide de la première hypothèse en prenant les vecteurs propres normalisés de $\mathbf{Y}'\mathbf{Y}$ et en multipliant la matrice qu'ils forment par $\sqrt{T}$, de tel façon que nous obtenons :

$$\hat{\mathbf{F}}/\sqrt{T} = PCA\left(\frac{1}{T}\mathbf{Y}'\mathbf{Y}, K\right)$$

Enfin, toujours en utilisant l'hypothèse de normalisation, on trouve naturellement un estimateur de $\mathbf{\Lambda}$ à partir de $\hat{\mathbf{F}}$:

$$\hat{\mathbf{\Lambda}} = \mathbf{Y}\hat{\mathbf{F}}/T$$

3.2 Modèle à facteurs semi-paramétrique

Afin de modéliser l'espace de projection dans le futur, nous utilisons une spécialisation du modèle factoriel appelée *modèle à facteurs semiparamétrique*. Dans ce format, les coefficients Λ sont supposés être des fonctions semiparamétriques des caractéristiques $\mathbf{X}$. L'équation reliant $\mathbf{Y}$ à $\mathbf{F}$ devient alors :

$$\mathbf{Y} = \{\mathbf{G}(\mathbf{X}) + \mathbf{\Gamma}\}\mathbf{F}' + \mathbf{U}$$

Où $\mathbf{\Gamma}$ correspond à une partie aléatoire non observable des coefficients et $\mathbf{G}(\mathbf{X})$ est une matrice $p \times K$ composée d'un ensemble de K fonctions non paramétriques $\mathbf{g}_k$ comme indiqué ci-dessous :

$$(\mathbf{G}(\mathbf{X}))_{ik} = \mathbf{g}_k(\mathbf{X}_i)$$

Les fonctions $(\mathbf{g}_k)_{1\leq k\leq K} : \mathbb{R}^d \to \mathbb{R}$ sont appelées fonctions génératrices des facteurs et sont celles qui introduisent la participation des caractéristiques $\mathbf{X}$ dans l'équation. Le terme $\mathbf{\Gamma}$ est responsable de la partie des coefficients non identifiable par $\mathbf{X}$ et est donc supposé indépendant de celle-ci.

Nous considérons également une autre spécialisation de ce modèle où chacune de ces fonctions génératrices multivariées est supposée additive dans chaque composante des vecteurs $\mathbf{X}_i$. Dans ce cas, nous considérons à présent un ensemble de $K \times d$ fonctions génératrices $\{g_{kl}\}_{1\leq k\leq K,\ 1\leq l\leq d} : \mathbb{R} \to \mathbb{R}$ de sorte que :

$$\mathbf{g}_k(\mathbf{X}_i) = \sum_{l=1}^{d} g_{kl}(X_{il})$$

Nous appellerons $\mathbf{Z}$ la matrice formée par ces $K \times d$ fonctions.

3.3 Méthode *Sieve*

Pour estimer les fonctions $\mathbf{g}_k$, on utilise la méthode *Sieve*. Cette technique consiste à essayer d'estimer, à partir d'un ensemble de fonctions bien choisies, chacune des composantes g_{kl}.

Soit $S = \{\phi_1(x), \phi_2(x), \dots\}$ une base de fonctions qui génère un espace linéaire dense sur l'espace des fonctions dont les composantes génératrices g_{kl} font partie. Ainsi, pour une précision d'approximation donnée J, on a pour chaque $1 \leq k \leq K$ et $1 \leq l \leq d$:

$$g_{kl}(X_{il}) = \sum_{j=1}^{J} b_{j,kl}\phi_j(X_{il}) + R_{kl}(X_{il})$$

Où $b_{j,kl}$ sont les coefficients *sieve* et R_{kl} sont les fonctions résiduelles qui donnent l'erreur d'approximation. En format matriciel, nous avons :

$$\mathbf{b}'_k = (b_{1,k_1}, \dots, b_{J,k_1}, \dots, b_{1,k_d}, \dots, b_{J,k_d}) \in \mathbb{R}^{Jd}$$
$$\phi(\mathbf{X}_i)' = (\phi_1(X_{i1}), \dots, \phi_J(X_{i1}), \dots, \phi_1(X_{id}), \dots, \phi_J(X_{id})) \in \mathbb{R}^{Jd}$$

Ce qui nous amène à la modélisation finale :

$$\mathbf{Y} = \{\mathbf{\Phi}(\mathbf{X})\mathbf{B} + \mathbf{\Gamma}\}\mathbf{F}' + \mathbf{R}(\mathbf{X})\mathbf{F}' + \mathbf{U}$$

Où $\mathbf{B} = (\mathbf{b}_k)_{1\leq k\leq K}$ et $\mathbf{\Phi}(\mathbf{X}) = (\phi(\mathbf{X}_i))'_{1\leq i\leq p}$

Dans l'article [8], certaines restrictions sont présentées pour garantir la précision de l'approximation des fonctions *sieve* et pour garantir que la parcelle d'erreur $\mathbf{R}$ est suffisamment petite. Ainsi, pour notre implémentation de cet algorithme, nous ne choisissons que des bases qui respectent ces restrictions.

3.4 P-PCA dans les modèles à facteurs

Maintenant que nous avons présenté le modèle aux facteurs spécialisés avec la méthode *sieve*, nous pouvons détailler la méthode de *Projected PCA* présentée dans [8].

Dans le P-PCA, nous projetons les valeurs de rendements attendus $\mathbf{Y}$ dans l'espace généré par les fonctions de tamisage appliquées à $\mathbf{X}$. La matrice de projection $\mathbf{P}$ est définie par :

$$\mathbf{P} = \mathbf{\Phi}(\mathbf{X})[\mathbf{\Phi}(\mathbf{X})'\mathbf{\Phi}(\mathbf{X})]^{-1}\mathbf{\Phi}(\mathbf{X})'$$

Après projection, nous pouvons écrire l'équation principale simplement comme suit :

$$\mathbf{PY} = \mathbf{\Phi}(\mathbf{X})\mathbf{BF}' + \tilde{\mathbf{E}}$$

Où le dernier terme, défini par l'expression :

$$\tilde{\mathbf{E}} = \mathbf{P}\boldsymbol{\Gamma}\mathbf{F}' + \mathbf{P}\mathbf{R}(\mathbf{X})\mathbf{F}' + \mathbf{P}\mathbf{U}$$

exprime l'erreur d'estimation. L'avantage de la P-PCA sur son prédécesseur vient du fait que ce terme est négligeable lorsque $\mathbf{U}$ et $\boldsymbol{\Gamma}$ sont orthogonaux à l'espace de projection et que les fonctions *sieve* approchent suffisamment bien les composantes des facteurs.

Nous pouvons considérer l'erreur $\tilde{\mathbf{E}}$ comme négligeable et procéder de la même manière qu'avec le PCA traditionnelle présentée précédemment. Dans ce cas, nous aurons les hypothèses d'identification suivantes :

$$\begin{cases} \frac{1}{T}\mathbf{F}'\mathbf{F} = \mathbf{I_k} \\ \boldsymbol{\Lambda}'\mathbf{P}\boldsymbol{\Lambda} \text{ est une matrice diagonale avec des entrées distinctes} \end{cases}$$

De manière analogue à l'ACP traditionnelle, nous trouvons l'estimateur suivant pour $\mathbf{F}$:

$$\hat{\mathbf{F}}/\sqrt{T} = PCA\left(\frac{1}{T}\mathbf{Y}'\mathbf{P}\mathbf{Y}, K\right)$$

L'estimateur de $\hat{\boldsymbol{\Lambda}}$ à partir de $\hat{\mathbf{F}}$ reste identique à celle de PCA et nous pouvons estimer de la même manière $\mathbf{G}(\mathbf{X}) \approx \boldsymbol{\Phi}(\mathbf{X})\mathbf{B}$:

$$\hat{\boldsymbol{\Lambda}} = \mathbf{Y}\hat{\mathbf{F}}/T$$
$$\hat{\mathbf{G}}(\mathbf{X}) = \mathbf{P}\mathbf{Y}\hat{\mathbf{F}}/T$$

Finalment, après avoir estimé $\mathbf{F}$, nous pouvons utiliser $\hat{\mathbf{F}}$ pour trouver les coefficients *sieve* dans l'équation :

$$\mathbf{P}\mathbf{Y} = \boldsymbol{\Phi}(\mathbf{X})\mathbf{B}\mathbf{F}' + \tilde{\mathbf{E}}$$

En multipliant l'équation ci-dessus à gauche par $[\boldsymbol{\Phi}(\mathbf{X})'\boldsymbol{\Phi}(\mathbf{X})]^{-\mathbf{1}}\boldsymbol{\Phi}(\mathbf{X})'$ et à droite par $\mathbf{F}$ nous avons que :

$$\hat{\mathbf{B}} = \frac{1}{T}[\boldsymbol{\Phi}(\mathbf{X})'\boldsymbol{\Phi}(\mathbf{X})]^{-\mathbf{1}}\boldsymbol{\Phi}(\mathbf{X})\mathbf{Y}\hat{\mathbf{F}}$$

Il faut remarquer que l'estimateur proposé par les auteurs n'est pas bien défini pour tous les $\mathbf{X}$ car la matrice $\boldsymbol{\Phi}(\mathbf{X})'\boldsymbol{\Phi}(\mathbf{X})$ peut ne pas être inversible.

Pour faire face à de telles situations, connues sous le nom de cas de multicolinéarité, nous préférons appliquer une régression Ridge sur $G(X)$ pour trouver les coefficients souhaités.

3.5 L'estimation de K

Jusqu'à présent, nous avons travaillé avec l'hypothèse que le nombre de facteurs **K** est connu. Mais dans la pratique, nous ne le connaissons pas et cela affecte profondément le comportement des estimations puisque cette variable détermine la taille de la matrice des facteurs qui est la base pour le reste des estimations.

Pour faire face à ce problème, nous devons également estimer le nombre de facteurs et nous avons choisi deux procédures pour ce faire. Dans les deux cas, l'estimation est effectuée au cours de l'étape de la PCA, au cours de laquelle les valeurs propres de $\mathbf{Y}'\mathbf{PY}$ sont triées par ordre décroissant.

3.5.1 • Méthode d'*Elbow*

La première est connue sous le nom de méthode d'*Elbow* [14] et est la manière classique de choisir le nombre de composantes dans PCA. Cette méthode consiste simplement à tracer les valeurs propres trouvées et à rechercher un point semblable à un coude dans la courbe.

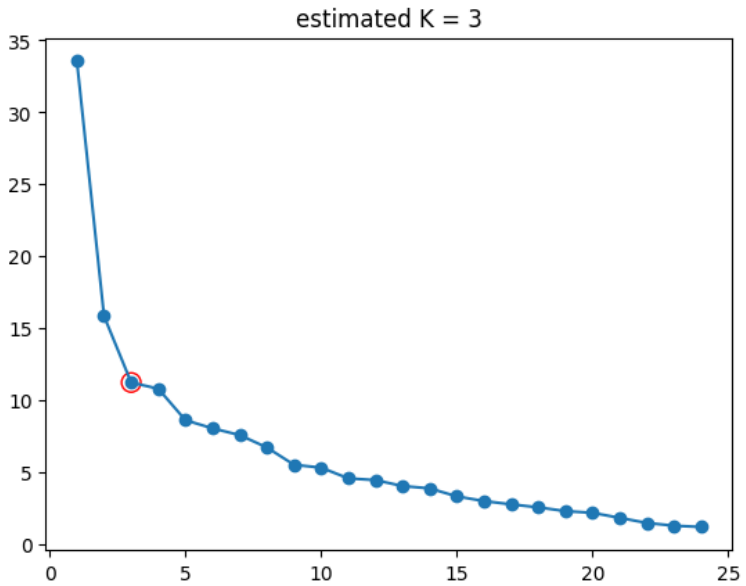


Figure 2 – Illustration de l'estimation de K par la méthode d'*Elbow*

Il s'agit d'une méthode totalement heuristique dont le principe consiste à essayer de trouver un petit nombre de facteurs, afin d'éviter les problèmes de dimensionnalité, tout en expliquant un maximum de variance, ce qui équivaudrait à choisir des valeurs propres de plus grande amplitude.

3.5.2 • Adjacent Ratio

La deuxième méthode a été introduite par Ahn et Horenstein [10], d'abord pour le PCA classique, mais elle peut également être étendue pour le *Projected-PCA* sous certaines conditions, comme cela a été observé dans [8].

Cette méthode alternative est basée sur des principes similaires à la précédente et consiste à sélectionner le plus grand rapport de valeurs propres adjacentes en triant la liste qu'elles forment par ordre décroissant. Plus formellement :

$$\hat{K} = \underset{k}{\operatorname{argmax}} \frac{\lambda_k(\mathbf{Y}'\mathbf{P}\mathbf{Y})}{\lambda_{k+1}(\mathbf{Y}'\mathbf{P}\mathbf{Y})}$$

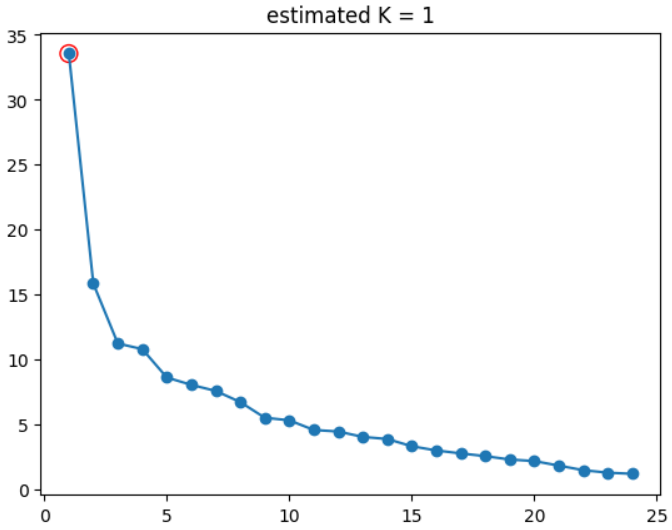


Figure 3 – Illustration de l'estimation de K par la méthode des rapports adjacents

3.6 Implementation

Pour mettre en œuvre cette méthode, nous avons utilisé le langage de programmation *Python*, avec les bibliothèques *sklearn* pour les métriques et les algorithmes d'apprentissage automatique, *matplotlib* pour les graphiques, *pandas* pour le traitement des données réelles et *numpy* pour les calculs mathématiques. Nous avons réalisé notre propre implémentation et l'implémentation avec les outils cités. En utilisant *sklearn*, nous avons généré deux classes de fonctions de base pour le sieve : *B-Spline*(cubique), en utilisant le *SplineTransformer* et *Polynomial*, en utilisant les *PolynomialFeatures*. Dans notre propre implémentation, nous avons généré les deux mêmes classes de fonctions de base et, en outre, la série de Fourier. Pour ce projet, nous avons utilisé la plateforme Xalpha comme environnement de programmation collaborative, afin que le tuteur puisse avoir accès à la progression du travail.

4
SIMULATION DES DONNÉES

Afin de démontrer la performance du modèle P-PCA et d'évaluer les hypothèses formulées à son sujet, nous avons étudié et mis en œuvre des modèles qui simulent le rendement attendu en tant que modèle à facteurs.

Par la suite, ces données simulées ont été utilisées pour vérifier l'erreur associée à la prédiction lorsque des modèles basés sur la régression linéaire sont appliqués à mesure que la nature non linéaire des fonctions caractéristiques augmente.

4.1 Méthodologie

Pour réaliser la simulation, deux modèles génératifs ont été étudiés et mis en œuvre, à savoir :

1. Simulation avec la méthode de Monte Carlo [7]
2. Simulation par autorégression vectorielle (VAR) [8]

Dans le premier modèle, nous utilisons la méthode de Monte Carlo pour simuler les caractéristiques des actifs au cours du temps, tandis que dans le deuxième nous considérons les caractéristiques comme des variables aléatoires i.i.d (indépendentes et identiquement distribuées). De plus, nous appliquons dans le deuxième modèle l'autorégression vectorielle pour définir les facteurs de notre modèle.

Dans les deux modèles mentionnés ci-dessus, le choix de la fonction caractéristique est arbitraire, pour que nous puissions la changer de façon à augmenter ou réduire son caractère linéaire, ce qui sera important pour analyser comment cette modification impacte la prédiction en utilisant les modèles qui sont basés sur la régression linéaire.

Après la mise en œuvre, pour évaluer si la simulation est cohérente avec un modèle factoriel, c'est-à-dire un modèle dans lequel le rendement attendu est corrélé aux caractéristiques, nous générons les données simulées et essayons de les prédire à l'aide des modèles **OLS** (*Ordinary Least Squares*) et **Random Forest**. De cette manière, il est possible de vérifier si les rendements simulés sont prévisibles à partir de leurs caractéristiques ou si une telle prédiction s'avère impossible.

4.2 Description des modèles

Nous allons maintenant décrire les modèles génératifs utilisés pour réaliser la simulation des données.

4.2.1 • Simulation avec la méthode de Monte Carlo

Dans cette partie, nous considérons que, pour chaque moment t considéré, les caractéristiques c_t sont des matrices de taille $N \times P_c$, où N est le nombre des actifs et P_c est le nombre des caractéristiques pour chaque actif. Nous allons simuler un modèle à 3 facteurs, où nous considérons les facteurs comme les trois premières caractéristiques. L'équation qui décrit les rendemments attendus est donné par :

$$r_{i,t+1} = g^*(c_{i1,t},\, c_{i2,t},\, c_{i3,t},\, x_t) + e_{i,t+1}$$

Avec :

$$\begin{cases} g^*(c_{i1,t},\, c_{i2,t},\, c_{i3,t},\, x_t) = (c_{i1,t},\, c_{i2,t},\, x_t \cdot c_{i3,t})(0,02\ 0,02\ 0,02)^T \\ e_{i,t+1} = (c_{i1,t},\, c_{i2,t},\, c_{i3,t})v_{t+1} + \epsilon_{i,t+1} \end{cases}$$

Nous nous amenons à l'explication de chaque terme et, en suite, à l'explication sur la génération des caractéristiques c_t.

— Ici, le terme $g^*(c_{i1,t},\, c_{i2,t},\, c_{i3,t},\, x_t)$ est la fonction qui met en corrélation les caractéristiques observées avec le rendement attendu. Nous utilisons également les séries temporelles x_t comme argument de cette fonction, où $x_0 \sim \mathcal{N}(0,\, 1)$ et, pour $t \geq 1$, x_t est définie par la récurrence :

$$x_t = \rho x_{t-1} + u_t, \quad \text{avec} \quad u_t \sim \mathcal{N}(0,\, 1-\rho^2) \text{ et } \rho = 0,95$$

Les paramètres ont été choisies pour garantir que x_t soit persistent avec le temps. Nous voyons donc que g^* donne une corrélation linéaire entre les caractéristiques et les rendemments attendus.

— Le terme $e_{i,t+1}$ exprime le bruit liée à la prédiction de chaque actif i au moment $t+1$ et il est une fonction des caractéristiques multiplié par v_{t+1} de taille 3×1 tel que $v_{t+1} \sim \mathcal{N}(0,\ 0,05^2 \times I_3)$, où I est la matrice identité plus les erreurs idiosyncratiques donnés par $\epsilon_{i,t+1} \sim t_5(0,\ 0,05^2)$, où t_5 est la Loi de Student de degré de liberté 5.

Enfin, les caractéristiques c_t sont génerées parmi la méthode de Monte Carlo : nous définissons le parámètre $\bar{c}_{ij,\,t}$ tel que, pour $t = 0$, $\bar{c}_{ij,\,0} \sim \mathcal{N}(0,\ 1)$ et pour $t \geq 1$ nous avons la récurrence :

$$\bar{c}_{ij,\,t} = \rho_j \bar{c}_{ij,\,t-1} + \epsilon_{ij,\,t} \quad \text{avec } \rho_j \sim \mathcal{U}[0.9,\ 1] \text{ et } \epsilon_{ij,\,t} \sim \mathcal{N}(0,\ 1-\rho_j^2)$$

À partir de ces parámètres, nous définissons chaque composant $c_{ij,\,t}$ par l'équation :

$$c_{ij,t} = \frac{2}{N+1} CSrank(\bar{c}_{ij,\,t}) - 1$$

Où $CSRank$ est la fonction Cross-Section rank. Donc, en générant les matrices de caractéristiques c_t, nous pouvons simuler les rendemments attendus avec un modèle à facteur.

4.2.2 • Simulation par VAR

Dans le modèle P-PCA, nous décrivons le rendement attendu $y_{i,t}$ pour l'entreprise i au moment t comme un modèle semi-paramétrique de ses caractéristiques. L'équation qui décrit cette rélation est donnée par :

$$\mathbf{Y} = \{\mathbf{G}(\mathbf{X}) + \mathbf{\Gamma}\}\mathbf{F}' + \mathbf{U}$$

Pour la suite de la simulation, nous avons considéré $\Gamma = 0$, de tel façon que l'expression devient simplement :

$$\mathbf{Y} = \mathbf{G}(\mathbf{X})\mathbf{F}' + \mathbf{U}$$

Nous nous amenons maintenant à l'explication de chaque terme et la manière dont il a été généré.

— La matrice des caractéristiques $\mathbf{X}$ de taille $p \times d$ est générée selon une distribution normale standard, c'est à dire, pour chaque élement x_{ij} de la matrice, nous avons $x_{ij} \sim \mathcal{N}(0,\, 1)$

— Pour construire la matrice $\mathbf{G}$ de taille $p \times K$, nous considérons l'ensemble de fonctions caractéristiques $\{g_j\}_{1 \le j \le K}$ et pour chaque ligne i, nous considérons le vecteur $\mathbf{X_i}$ tel que :

$$\mathbf{X_i} = \begin{bmatrix} X_{i1} & X_{i2} & \cdots & X_{id} \end{bmatrix}$$

De tel façon que la matrice $\mathbf{G}$ est définie par $\mathbf{G} = \{g_j(X_i)\}_{1 \le i \le p;\, 1 \le j \le K}$, c'est à dire :

$$\mathbf{G}(\mathbf{X}) = \begin{bmatrix} g_1(X_1) & g_2(X_1) & \cdots & g_K(X_1) \\ g_1(X_2) & g_2(X_2) & \cdots & g_K(X_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(X_d) & g_2(X_d) & \cdots & g_K(X_d) \end{bmatrix}$$

— la matrice des facteurs $\mathbf{F}$ de taille $T \times K$ est décrite par :

$$\mathbf{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_T \end{bmatrix} \quad \text{avec} \quad \begin{cases} f_0 \sim \mathcal{N}(0,\, I_K) \\ f_t = \mathbf{A} f_{t-1} + \varepsilon_t \end{cases}$$

Où chaque f_t est un vecteur ligne, défini par **l'auto régression vectorielle**. En plus, nous avons ici la matrice $\mathbf{A} = (a_{ij})_{1 \le i,j \le K}$ tel que $a_{ij} \sim \mathcal{U}(-0,3,\ 0,3)$ et ε_t et un vecteur de dimension K tel que $\varepsilon_t \sim \mathcal{N}(0,\, \mathbf{I_k})$, où $\mathbf{I_k}$ est la matrice identité de taille $K \times K$.

— La matrice $\mathbf{U}$ qui exprime l'erreur idiosyncratique est générée à partir des matrices $\mathbf{\Lambda} = \mathbf{G}(\mathbf{X}) + \mathbf{\Gamma}$ et $\mathbf{F}$. Dans cette simulation, comme $\mathbf{\Gamma} = 0$, nous avons que $\mathbf{\Lambda} = \mathbf{G}(\mathbf{x})$. Pour générer $\mathbf{U}$, nous calculons la variance de la matrice $\mathbf{\Lambda F'} = \mathbf{G}(\mathbf{X})\mathbf{F'}$ et nous définissons $\mathbf{U} = \{u_{ij}\}_{1 \leq i \leq p,\, 1 \leq j \leq T}$ de tel façon que :

$$u_{ij} \sim \mathcal{N}\left(0,\, \mathrm{Var}\left(\mathbf{G}(\mathbf{X})\mathbf{F'}\right)\right)$$

Finalement, il faut remarquer que nous avons toujours les conditions de normalization sur les matrices $\mathbf{F}$ et $\mathbf{\Lambda} = \mathbf{G}(\mathbf{x})$, qui sont données par :

$$\begin{cases} \frac{1}{T}\mathbf{F'F} = \mathbf{I_k} \\ \mathbf{\Lambda' P \Lambda} \text{ est une matrice diagonal avec des entrées distinctes} \end{cases}$$

4.3 Évaluation des simulations

Pour évaluer la prévisibilité des simulations et vérifier s'il existe une corrélation entre les caractéristiques et les rendements attendus simulés, une validation hors échantillon (ou validation croisée) est effectuée pour chacun des modèles.

Pour chaque modèle de simulation et chaque modèle de prédiction, nous analysons les valeurs de R^2 (coefficient de détermination), MAE (*Mean Absolute Error*) et MSE (*Mean Squared Error*).

4.3.1 • Simulation avec la méthode de Monte Carlo

Pour la simulation avec la méthode de Monte Carlo, nous avons utilisé la fonction caractéristique qui est donnée par l'expression :

$$g^*(c_{i1,t},\, c_{i2,t},\, c_{i3,t},\, x_t) = (c_{i1,t},\, c_{i2,t},\, x_t \cdot c_{i3,t})(0,02\ \ 0,02\ \ 0,02)^T$$

C'est à dire, Nous attendons que la corrélation entre les rendemments et les caractéristiques soient linéaires. Les résultats de la validation croisée pour ce modèle en utilisant la régression OLS et la régression *Random Forest* sont présentés dans le tableau ci-dessous :

Modèle	R^2	MAE	MSE
Régression OLS	-0,0313	0,0233	0,076
Random Forest	-0,007	0,227	0,074

Dans le langage de programmation *Python*, la valeur négative du coefficient de détermination R^2 indique que les données Y (rendemments attendus) sont imprévisibles à partir de X (caractéristiques des actifs). Les résultats ci-dessus nous permettent donc de conclure que la simulation par la méthode de Monte Carlo décrite ci-dessus n'a pas réussi à bien décrire le modèle à facteurs désiré, car même si la fonction caractéristique est linéaire, la régression *OLS* n'a pas pu être bien effectuée.

4.3.2 • Simulation par VAR

Pour cette simulation, nous considérons les valeurs suivantes pour les paramètres :

Paramètre	d	K	p	T
Valeur	3	1	20	10

Et les 3 fonctions caractéristiques que nous avons utilisé sont :

$$\begin{cases} g_1(x) = x \\ g_2(x) = x^2 - 1 \\ g_3(x) = x^3 - 2x \end{cases}$$

De tel façon que la corrélation entre les rendemments attendus et les facteurs n'est pas linéaire. Pour évaluer ce modèle de simulation, nous nous amenons dans un premier temps au cas où l'erreur idiosyncratique est nulle, c'est à dire, $U = 0$. Dans un deuxième moment, nous ajoutons cet erreur comme décrit dans la sous-section précedente.

Les résultats obtenus pour le cas où l'erreur idiosyncratique est nulle sont :

Modèle	R^2	MAE	MSE
Régression OLS	0,252	13,090	2,032
Random Forest	0,933	1,164	8,944

Pour le deuxième cas, nous avons les résultats :

Modèle	R^2	MAE	MSE
Régression OLS	0,133	5,787	1,406
Random Forest	0,815	1,211	0,533

Nous faisons quelques commentaires sur ces résultats obtenus :

— Quand nous considérons le cas sans bruit, nous observons que la simulation par VAR a bien réussi à décrire le modèle à facteurs, étant donné que la prédiction peut être faite, soit par la régression *OLS*, soit par la régression *Random Forest*.

— Comme nous pouvons attendre, la qualité de la prédiction diminue dans le second cas, après l'ajout du bruit lié à l'erreur idiosyncrasique. Par contre, même en considérant la régression *OLS* dans le deuxième cas, nous observons que la simulation continue prévisible à un certain niveau, de façon que la simulation a été bien réalisée.

— Le coefficient de détermination R^2 est beaucoup plus important dans la régression *Random Forest* que dans la régression *OLS* et les valeurs de MAE et MSE sont moins importantes dans la régression *Random Forest* que dans la régression *OLS* aussi. De tels faits peuvent être expliqués par le fait que la corrélation entre Y et X n'est pas linéaire, comme nous pouvons remarquer par le caractére non-linéaire des fonctions $g_2(x)$ et $g_3(x)$.

5
RÉSULTATS

5.1 Influence de la linéarité dans l'erreur de prédiction

Le premier résultat obtenu par notre groupe est relatif aux erreurs de prédiction lorsque le caractère linéaire des fonctions caractéristiques diminue : comme mentionné dans l'introduction de ce rapport, de nombreux modèles de prédiction tels que la régression *OLS* sont assez sensibles au type de corrélation entre les caractéristiques et les rendements attendus. Ainsi, en utilisant la simulation VAR, nous vérifions comment cette variation de la corrélation affecte l'erreur de prédiction pour différents modèles de prédiction.

Les valeurs des paramètres utilisés pour la simulation des données sont indiquées dans le tableau ci-dessous :

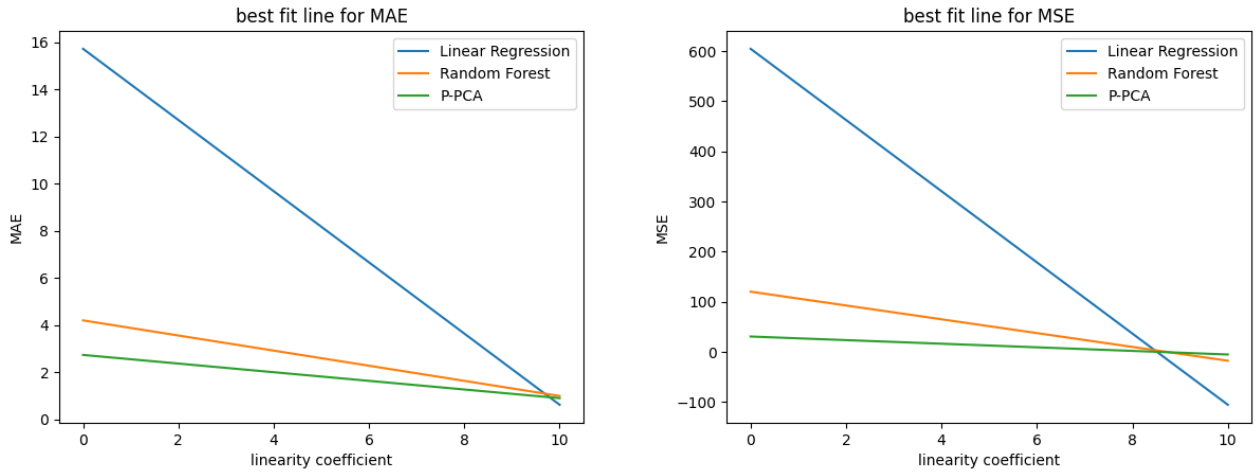
Paramètre	d	p	T	K
Valeur	1	250	10	3

Cependant, contrairement à ce qui a été fait jusqu'à présent, nous utilisons ici 3 fonctions caractéristiques qui dépendent d'un autre paramètre C que nous appelons **coefficient de linéarité** et qui varie entre les valeurs 0 et 10. Pour chaque valeur de C, nous considerons les fonctions caractéristiques suivantes :

$$\begin{cases} g_1(x) = Cx \\ g_2(x) = (10 - C)x^2 - C \\ g_3(x) = (10 - C)x^3 - 2Cx \end{cases}$$

Ainsi, nous constatons que lorsque $C = 0$, les fonctions sont totalement non linéaires et que pour $C = 10$, toutes les fonctions sont linéaires, le caractère linéaire des fonctions variant pour les valeurs intermédiaires entre 0 et 10.

Nous considérons les valeurs *MAE* et *MSE* pour les modèles de régression *OLS*, *Random Forest* et pour le *Projected-PCA* et pour chaque valeur entière de C, nous effectuons 50 itérations et prenons la moyenne des erreurs. Ensuite, nous calculons et traçons la droite qui se rapproche le plus de ces valeurs, ce qui nous permet d'obtenir les graphiques ci-dessous :



(a) graphique de *MAE* en fonction de C

(b) graphique de *MSE* en fonction de C

Figure 4 – Graphiques des erreurs associées aux prédictions en fonction du coefficient de linéarité

En ce qui concerne les résultats obtenus, nous pouvons observer que :

— Comme est attendu pour la régression *OLS*, l'erreur de prédiction diminue considérablement lorsque le caractère linéaire des fonctions caractéristiques devient plus évident, étant donné que cette prédiction suppose une corrélation linéaire entre le rendement attendu et ses caractéristique.

— Bien qu'il y ait une réduction des erreurs de prédiction pour la régression *Random Forest* et *projected-PCA*, nous observons que ces variations sont très faibles par rapport à la régression *OLS*. Ce phénomène peut s'expliquer par le fait que ces méthodes prédictives ne font aucune hypothèse quant au type de corrélation entre le rendement attendu et ses caractéristiques, de sorte qu'une simplification du type de corrélation améliore la prédiction, mais ne change pas dans la même proportion que pour la régression par les *OLS*.

5.2 Estimation de K

Nous nous amenons maintenant à l'évaluation des performances des deux méthodes d'estimation K étudiées. Pour celui-là, nous avons utilisé la simulation par VAR pour générer des données avec plusieurs quantités de facteurs et de fonctions génératrices linéaires et non-linéaires afin de comprendre comment les deux méthodes fonctionnent dans ces situations. Les fonctions utilisées sont les suivantes :

$$\begin{cases} g_1(x) = x \\ g_2(x) = x^2 \\ g_3(x) = x^3 \\ g_4(x) = x^4 \\ g_{sinc}(x) = \frac{\sin(3\pi x)}{3\pi x} \\ g_{cos}(x) = \cos(x) \\ g_{sen}(x) = \sin(x) \\ g_{tan}(x) = \tan(x) \\ g_{exp}(x) = \exp(x) \end{cases}$$

Dans un premier temps, nous réalisons les simulations en utilisant seulement une caractéristique, c'est à dire, avec paramètre $d = 1$. Nous présentons sous forme de tableau les fonctions caractéristiques utilisées, la valeur réelle de K et les valeurs de K estimées par chacun des méthodes :

Fonctions utilisées	K réelle	K_{elbow}	K_{adj}
g_1, g_2, g_3	3	3	3
g_1, g_2, g_3, g_{sinc}	4	3	4
g_1, g_2, g_3, g_{sinc}, g_{cos}	5	2	5
g_{sin}, g_{sinc}, g_{cos}	3	3	3
g_1, g_{cos}, g_{exp}	3	2	3
g_1, g_{cos}, g_{exp}, g_{sinc}	4	2	4
g_1, g_2, g_3, g_{sinc}, g_{cos}, g_{exp}	6	3	5

Ces résultats montrent que la méthode proposée dans l'article [8] est plus performante que la méthode d'*Elbow*, principalement lorsque le nombre de facteurs augmente. Ce n'est que dans le dernier test que le résultat des deux méthodes ne correspond pas au nombre correct.

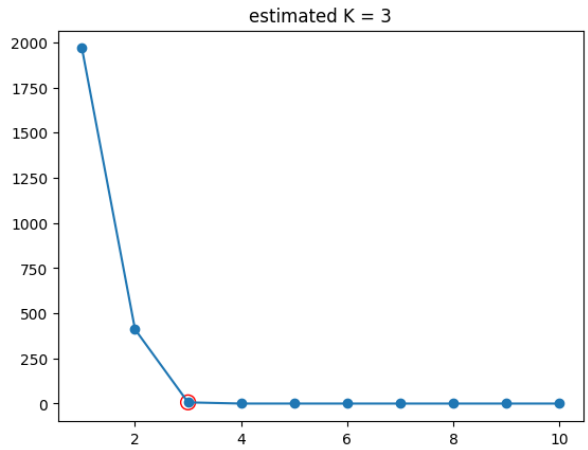
Nous avons également testé les cas $d = 2$ et $d = 3$, pour les configurations suivantes des fonctions génératrices :

$$\mathbf{Z_1} = \begin{bmatrix} g_3 & g_2 \\ g_1 & g_4 \end{bmatrix}, \quad \mathbf{Z_2} = \begin{bmatrix} g_{mix} & g_{\cos} \\ g_{\exp} & g_{arctg} \\ g_1 & g_2 \end{bmatrix}, \quad \mathbf{Z_3} = \begin{bmatrix} g_{\sin} & g_{\cos} & g_{\tan} \\ g_1 & g_2 & g_3 \end{bmatrix}$$

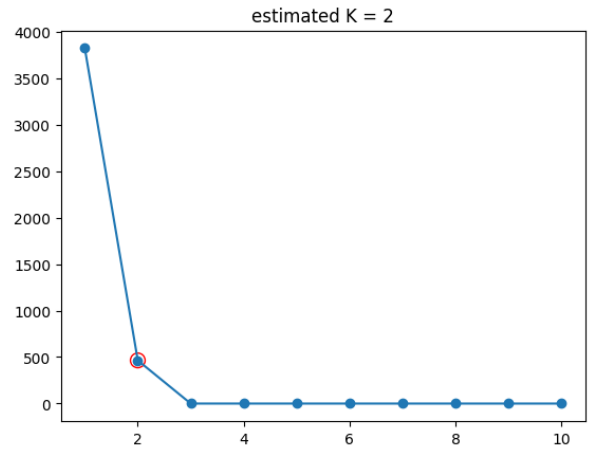
Nous présentons sous forme de tableau les configurations utilisés, la valeur réelle de K et les valeurs de K estimées par chacun des méthodes, bien comme la valeur de d utilisée :

Configuration utilisé	d	K réelle	K_{elbow}	K_{adj}
$\mathbf{Z_1}$	2	2	2	2
$\mathbf{Z_2}$	2	3	3	3
$\mathbf{Z_3}$	3	2	3	2

La méthode proposée a permis d'obtenir presque toujours le bon nombre de facteurs pour les 3 valeurs de d utilisées, alors que la méthode *Elbow* nous donne inconditionnellement 2 ou 3 facteurs. Une étude plus approfondie serait nécessaire pour pouvoir évaluer plus profondément l'efficacité des deux méthodes dans des situations avec $d > 1$.



(a) Estimation de K pour $\mathbf{Z_2}$



(b) Estimation de K pour $\mathbf{Z_3}$

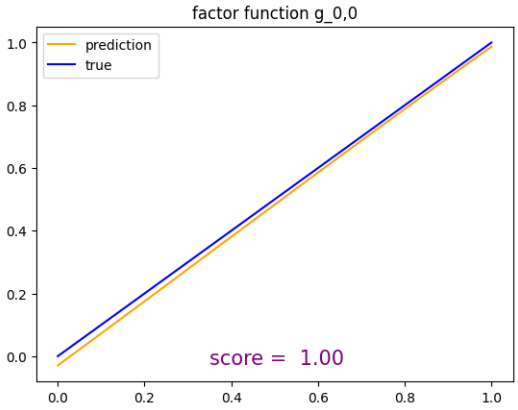
Figure 5 – Illustrations relatives à l'estimation de K pour $d > 1$

5.3 Estimation des courbes caractéristiques

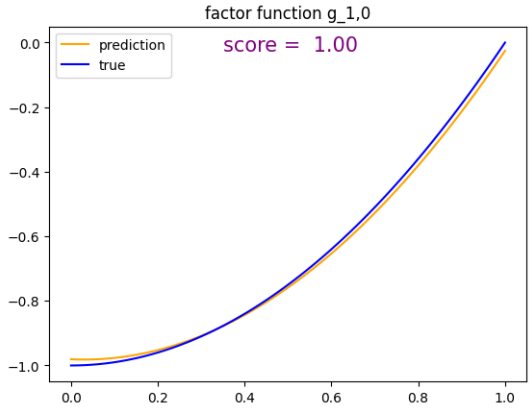
À partir des données simulées avec la méthode VAR, nous avons mis en œuvre la méthode P-PCA améliorée par la régression Ridge pour essayer de retrouver la forme des courbes génératrices. Nous réalisons ici deux tests : le premier avec $K = 3$ facteurs et $d = 1$ caractéristique et le deuxième avec 2 facteurs et 2 caractéristiques, c'est à dire, $K = d = 2$. Avec ces tests, nous pouvons observer comment le nombre des caractéristiques change la performance de l'estimation des couber des fonctions caractéristiques.

5.3.1 • Premier test : $d = 1$ et $K = 3$

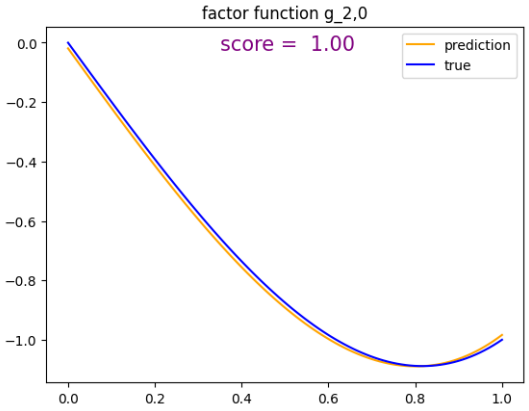
Dans un premier temps, nous considérons $K = 3$ et $d = 1$ et nous utilisons les mêmes fonctions que celles citées dans la section sur l'évaluation des VAR en utilisant des splines cubiques avec un degré d'approximation *sieve* $J = 6$. les graphiques obtenus pour ces estimations sont illustrés ci-dessous :



(a) Graphique de l'estimation de $g_1(x) = x$



(b) Graphique de l'estimation de $g_2(x) = x^2 - 1$



(c) Graphique de l'estimation de $g_3(x) = x^3 - 2x$

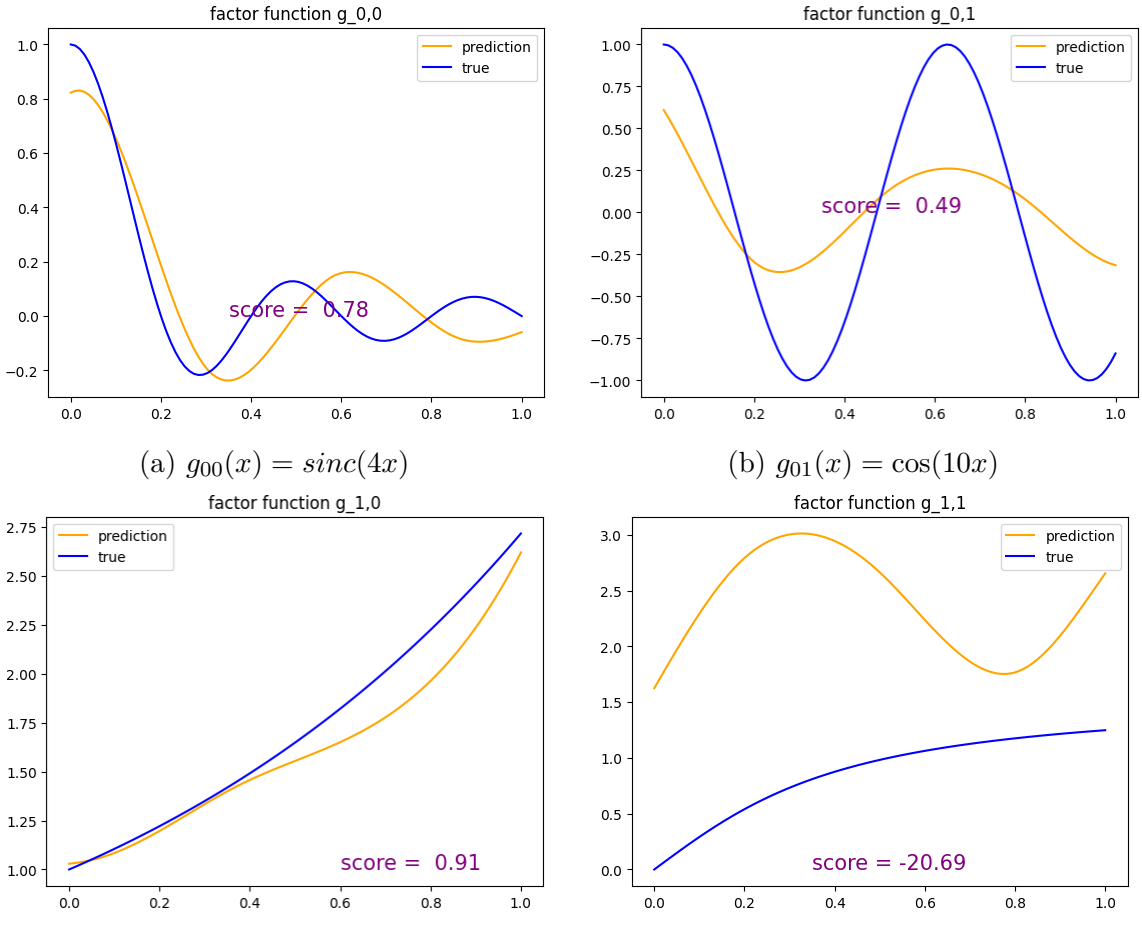
Figure 6 – Estimation des fonctions caractéristiques pour $K = 3$ et $d = 1$

5.3.2 • Deuxième test : $K = d = 2$

Un deuxième test plus complexe a été effectué, en considérant $K = d = 2$, c'est-à-dire en considérant un plus grand nombre de caractéristiques. En plus, nous avons utilisé comme fonctions caractéristiques des fonctions qui ne sont pas des polynômes afin d'analyser comment la prédiction de telles courbes est effectuée, de sorte que les fonctions considérées sont :

$$\mathbf{Z}(x) = \begin{bmatrix} sinc(4x) & \cos(10x) \\ \exp(x) & arctg(3x) \end{bmatrix}$$

Pour ces fonctions caractéristiques, les résultats obtenus pour une base également de splines cubiques avec un degré d'approximation $J = 20$ sont les suivants :



(a) $g_{00}(x) = sinc(4x)$ (b) $g_{01}(x) = \cos(10x)$

(c) $g_{10}(x) = \exp(x)$ (d) $g_{11}(x) = arctg(x)$

Figure 7 – Estimation des fonctions caractéristiques pour $K = 2$ et $d = 2$

5.3.3 • Analyse des résultats

À partir des résultats obtenus ci-dessus, nous pouvons faire les observations suivantes concernant l'estimation des courbes :

— Nous observons que dans le premier cas le résultat a été assez satisfactoire et les fonctions on été trouvées avec un dégré d'approximation quasi-maximal ($R^2 \approx 1$). L'explication pour cette performance peut être expliqué par le type de fonction caractéristique utilisé, étant tous des polynomes, et aussi à cause du nombre des caractéristiques $d = 1$.

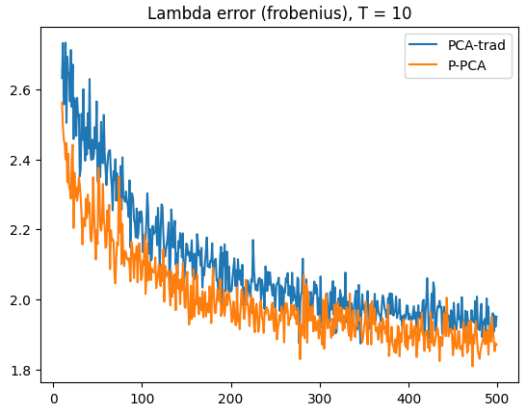
— Dans le deuxième cas nous contatons par contre que quand les dérivées des fonction varie trop rapidement ou quand les fonctions associé au même facteur sont trop différentes, la méthode n'est pas autant précise pour estimer les fonctions de départ, à l'exemple clair de l'*arctg* avec un score négatif.

En conclusion, nous pouvouns observer que deux aspects influencent de façon plus important l'estimation des courbes des fonctions caractéristiques : le type de fonction qui décrit les fonctions caractéristiques (nous avons une bonne estimation pour les courbes polynomiales, une performance moins importante pour les autres types de fonction) et le nombre de caractéristiques qui augmentent l'erreur d'estimation.

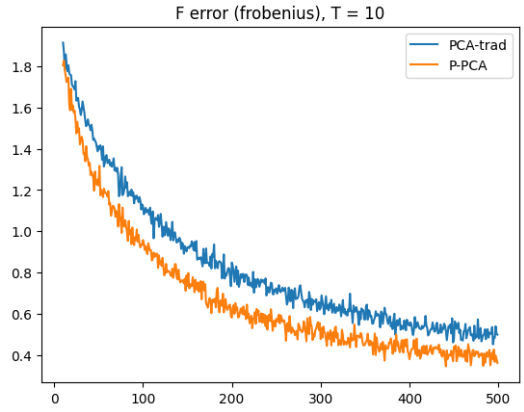
5.4 Comparaison : PCA et P-PCA

Pour effectuer la comparaison entre les méthodes PCA et P-PCA, nous avons analysé les erreurs associées aux estimations de $\mathbf{F}$ et $\mathbf{\Lambda}$ pour les normes de Frobenius et pour la norme sup en considérant les valeurs de $T = 10$ et $T = 50$. Pour cette comparaison, nous avons utilisé les résultats de simulation obtenus par la méthode VAR de manière analogue à ce qui a été fait et à son évaluation.

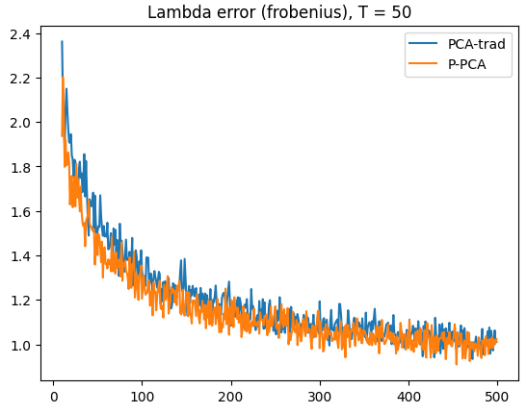
Pour ce test, pour chaque p compris entre 10 et 500, nous générons les données 500 fois et traçons la moyenne des erreurs, de sorte que nous obtenons les graphiques ci-dessous :



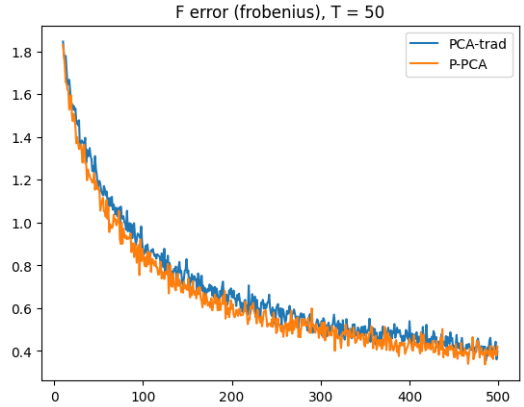
(a) Erreur de Λ pour $T = 10$



(b) Erreur de F pour $T = 10$



(c) Erreur de Λ pour $T = 50$



(d) Erreur de F pour $T = 50$

Figure 8 – Graphiques des erreurs de Λ et F pour $T = 10$ et $T = 50$

Nous pouvons observer que pour $T = 10$, la différence de performance de P-PCA par rapport à PCA classique pour les deux estimations est plus importante que dans le cas $T = 50$. Dans les deux cas, le P-PCA est plus performante que le PCA méthode classique et la meilleure performance de P-PCA pour le cas $T = 10$ confirme sa raison d'être dans les situations de *HDLSS*, comme mentionné dans l'introduction. Le résultat obtenu est en accord avec celui présenté dans l'article principal, ce qui valide bien notre mise en œuvre.

5.5 Données réelles

Nous disposons d'un ensemble de données fourni par notre tuteur avec 180 actions et 40 caractéristiques observées mensuellement au cours de la période allant du 31/12/1999 au 31/07/2022, soit un total de 271 observations. En utilisant la paramétrisation adoptée, nous aurions alors :

Paramètre	d	p	T_{max}	K
Valeur	40	180	271	-

K étant un paramètre latent à estimer.

Pour évaluer les performances de la méthode P-PCA dans cet ensemble de données, nous avons pris des fenêtres temporelles de 30 mois et calculé la moyenne des 40 caractéristiques dans chaque intervalle de temps. De cette manière, les données sont adaptées au format accepté par le modèle cible, c'est-à-dire que $\mathbf{X}$ doit être indépendant du temps.

Nous avons ensuite exécuté des partitions aléatoires d'entraînement et de test avec les données 100 fois et dans un rapport de 3 : 1 en prennant la moyenne pour calculer les résultats. Pour chaque partition, nous avons mesuré l'erreur de prédiction du modèle. Plus précisément, une étude transversale des données (ou *cross-section analysis*) a été réalisée. Chaque séparation est faite entre les p stocks dont une partie est utilisée pour estimer $\mathbf{F}$ et les coefficients *sieve* $\mathbf{B}$, et une partie comme comparables pour le calcul de l'erreur.

Après quelques tests initiaux, nous avons détecté que $K = 1$ était une bonne approximation lorsque nous considérons les erreurs quadratiques et absolues, ce qui, par coïncidence, est également la valeur proposée par la méthode des ratios adjacents.

À titre de référence, après avoir fixé $\hat{K} = 1$, nous effectuons cette même procédure de répartition avec les modèles de régression linéaire et de random forest, en calculant les erreurs correspondantes.

Pour les 30 premiers mois de l'ensemble de données, nous obtenons les résultats suivants pour la moyenne des erreurs obtenues :

Modèle	MAE	MSE
Régression *OLS*	0,76	1,12
Random Forest	0,66	0,84
P-PCA	0,75	1,18

Les valeurs de retour étant très importantes par rapport aux valeurs de retour dans les données réelles, nous avons également étudié le score R^2 avec la même *rolling average* :

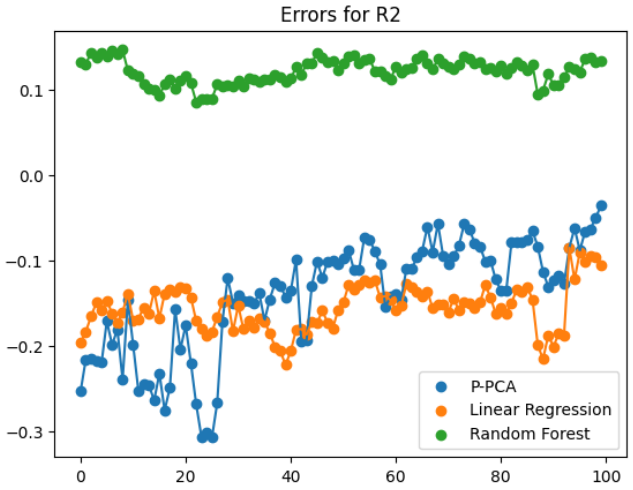


Figure 9 – Moyenne de R^2 sur une fenêtre glissante pour les 100 premiers mois

Il s'ensuit que la moyenne des caractéristiques sous une fenêtre glissante rend le nouvelles données ainsi générées insuffisamment corrélées, ce qui rend l'application de la P-PCA dans ce format irréalisable.

En outre, nous constatons que les données sont encore moins corrélées entre avril 2011 (mois 130) et juin 2016 (mois 180), comme le montre le graphique suivant :

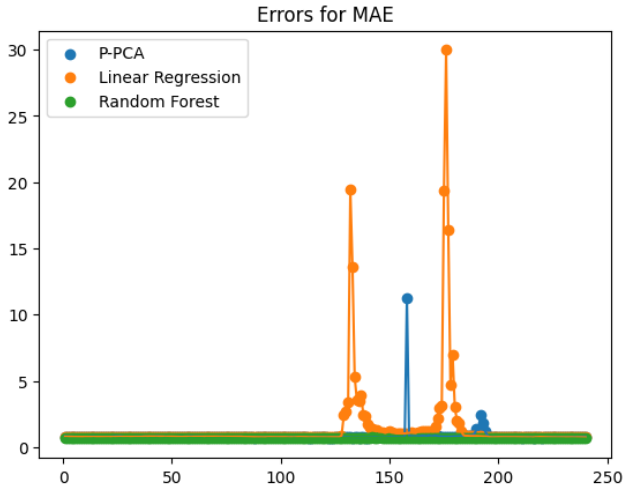


Figure 10 – Moyenne absolue de l'erreur sur une fenêtre glissante pour le jeu de données entier

6 CONCLUSION

Dans cette étude nous avons approfondi notre compréhension de la théorie de l'analyse en facteurs et appris à simuler des données pour mettre en œuvre et valider avec succès le modèle cible que nous avons étudié.

En utilisant une méthode de régression plus sophistiquée que celle proposée dans l'article de référence, nous avons réussi à récupérer des courbes génératrices pour diverses classes de fonctions. Nous avons également confirmé la meilleure performance de P-PCA par rapport à PCA traditionnel dans des situations de haute dimension et faible taille d'échantillon (HDLSS), ce qui sert de validation à notre propre mise en œuvre ainsi qu'à la robustesse du modèle P-PCA.

Nous avons exploré comment la non-linéarité entre les caractéristiques et le rendement affecte les performances des modèles populaires et avons observé comment P-PCA se compare avec ceux-ci en utilisant des données simulées en analyse transversale (ou *cross-section analysis*). Nous avons également montré que la méthode de la méthode du coude a une performance nettement inférieure à celle de la méthode des ratios adjacents. Enfin, avec les données réelles, nous avons testé l'hypothèse selon laquelle la moyenne des caractéristiques dans une fenêtre de temps serait applicable à la méthode P-PCA et avons conclu que la perte de données lors de la moyenne était suffisamment importante pour rendre les valeurs de rendement presque imprévisibles.

Pour des améliorations potentielles, nous suggérons une étude plus approfondie de la performance de la méthode d'estimation des facteurs pour $d > 1$, ainsi que l'utilisation d'autres fonctions de base de sieve et une étude de l'influence de ce choix sur la performance générale du modèle. En plus, étant donné que l'hypothèse selon laquelle les covariables sont indépendantes du temps peut être considérée comme assez restrictive pour les applications en finance, nous suggérons également, comme amélioration potentielle de cette méthode, de l'étudier dans des hypothèses moins restrictives où les covariables peuvent varier avec le temps.

7
RÉFÉRENCES BIBLIOGRAPHIQUES

1. Fama, E.F., and French, K.R. (1992), The Cross-Section of Expected Stock Returns, Journal of Finance, 47(2), pp. 427-465.
2. Fama, E.F., and French, K.R. (1993), Common Risk Factors in the Returns on Stocks and Bonds, Journal of Financial Economics, 33(1), pp. 3-56.
3. Fama, E.F., and French, K.R. (1998), Value versus Growth : The International Evidence, Journal of Finance, 53(6), pp. 1975-1999.
4. Fama, E.F., and French, K.R. (2012), Size, Value, and Momentum in International Stock Returns, Journal of Financial Economics, 105(3), pp. 457-472.
5. Giglio, Stefano and Kelly, Bryan T. and Xiu, Dacheng, Factor Models, Machine Learning, and Asset Pricing (October 15, 2021).
6. Hastie, T., Tibshirani, R. and Friedman, J. (2001), The Elements of Statistical Learning, Springer, New York.
7. Gu S, Kelly B, Xiu D. (2020), Empirical asset pricing via machine learning. Review of Financial Studies, 33(5), pp. 2223–2273
8. Fan, J., Liao, Y., and Wang, W. (2016), Projected Principal Component Analysis In Factor Models, The Annals of Statistics.
9. Bai, J. and Ng, S. (2002), Determining The Number Of Factors In Approximate Factor Models, Econometrica.
10. Ahn, S. C. and Horenstein, A. R. (2013). Eigenvalue ratio test for the number of factors. Econometrica 81 1203–1227
11. Lam, C. and Yao, Q. (2012). Factor modeling for high-dimensional time series : Inference for the number of factors. Ann. Statist. 40 694–726
12. Yata, K. and Aoshima, M. (2010). Effective PCA for high-dimension, low-sample-size data with singular value decomposition of cross data matrix, Journal of Multivariate Analysis 101 2060-2077
13. Jushan Bai, Serena Ng. (2013), Principal components estimation and identification of static factors, Journal of Econometrics, Volume 176, Issue 1
14. AlJanahi, Aisha & Danielsen, Mark & Dunbar, Cynthia. (2018). An Introduction to the Analysis of Single-Cell RNA-Sequencing Data. Molecular Therapy - Methods and Clinical Development. 10. 189-196. 10.1016/j.omtm.2018.07.003.
15. Fama, Eugene, F., and Kenneth R. French. 2004. "The Capital Asset Pricing Model : Theory and Evidence." Journal of Economic Perspectives, 18 (3) : 25-46.