

Conception de stratégies de *trading* quantitatif à travers la théorie des *rough paths signature*

Alphonse EBROTIE*, 16 mars - 19 mai 2024.

Introduction

La mise en place de stratégies de trading algorithmique est un aspect fondamental en finance quantitative et fait l'objet de plus en plus de recherches depuis ces dernières décennies. L'élaboration d'une stratégie de trading peut être divisée en deux modules : l'extraction de l'alpha et l'allocation du risque associé. Les méthodes d'extraction de l'alpha dépendent de l'horizon ou de la fréquence des transactions et sont souvent obtenues grâce à des techniques statistiques, qui peuvent ensuite être incorporées dans un modèle paramétrique. Parallèlement, l'allocation du risque associé dépend généralement du choix du modèle et du critère d'objectif ; par convention, des méthodes d'optimisation sont déployées pour trouver une transformation de la dynamique signal-actif qui maximise la fonction d'utilité choisie. En introduisant son papier [1], Markowitz propose une solution statique au problème d'optimisation de portefeuille fondé sur le critère rendement-risque. Ses travaux constituent dès lors, le début de la Théorie Moderne de l'optimisation de portefeuilles.

Cependant, la majorité des modèles proposés par cette théorie moderne du portefeuille reposent sur des hypothèses de marché trop simples, qui représentent mal la réalité, comme la gaussianité des rendements ou la markovianité des signaux de prix.

De plus, ces modèles ne parviennent souvent pas à capturer les faits stylisés présentés par les séries chronologiques financières [2] et ne prennent en compte que les facteurs traditionnels. C'est le prix à payer pour disposer de formules claires et souvent fermées pour les portefeuilles optimaux.

D'autre part, l'essor récent du *machine learning* et de l'informatique haute performance a incité de nombreux traders à privilégier une approche de l'investissement dite *data-driven*. Sous ce nouveau paradigme, on fait un minimum d'hypothèses ou pas d'hypothèses sur la dynamique du marché et on priorise fortement l'utilisation des données, en particulier les séries chronologiques financières.

Les algorithmes de trading construits avec le *machine learning* sont souvent plus sophistiqués et plus difficilement interpretable que les modèles traditionnels. En effet, ils peuvent ainsi capturer des signaux plus complexes mais, contrairement aux modèles plus classiques, ils peuvent rarement être expliqués par des formules fermées.

* alphonseebrotie@gmail.com

Ainsi , selon le paradigme sous lequel on se place , il y a un compromis entre les modèles classiques qui offrent des formules fermées mais ne sont pas capables de modéliser les marchés avec précision ou de façon plus réaliste et les approches dites *data-driven* fondées , sur les données qui peuvent capturer des signaux complexes mais n'offrent pas d'explication théorique claire (en terme de formule fermée).

Dans ce travail , il sera question d'explorer le potentiel de la théorie des *signature* de trajectoires afin de concevoir des stratégies de trading en tirant profit des avantages des deux mondes (*data-driven* et modèles offrant des formules fermées) par l'étude du papier de recherche [3] intitulé : *Signature Trading : A Path Dependent Extension of the Mean-Variance Framework with Exogenous Signals* de Fütter et al. Dans leurs travaux , les auteurs proposent d'utiliser le concept de signature afin de capturer des signaux complexes, tout en offrant des formules fermées pour le portefeuille optimal , ce qui est un bon compromis . Le modèle introduit incorpore aussi des dépendances dans la dynamique des actifs en plus de permettre un contrôle implicite du *drawdown*.

Table des matières

1	Le concept de signature	3
1.1	Eléments de contexte	3
1.2	Préliminaires d'algèbre tensorielle	3
1.3	Notion de signature	5
1.3.1	Quelques intuitions	6
1.3.2	Propriétés	7
1.3.3	Transformation de trajectoires	9
1.4	Quelques domaines d'utilisation de la signature	11
2	Signature trading	12
2.1	Cadre de modélisation	12
2.2	Algorithme sig-trader	16
3	Quelques Applications	17
3.1	Cas Markowitz vs Signature trading	17
3.2	Stratégie <i>Mean-reversion</i>	19
3.3	Stratégie de pairs trading	21
3.4	Prise en compte de signaux exogènes	24
	Conclusion	26

1 Le concept de signature

1.1 Éléments de contexte

Dans le papier de recherche , l'idée principale est d'utiliser la signature des trajectoires afin de construire des signaux de trading . En reprenant certaines notations du papier original , considérons que nous *tradons* un seul actif qu'on note X sans informations exogènes. Notre position notée ξ_t à tout instant t par rapport à l'actif n'est rien d'autre qu'une fonction de l'information (le prix de l'actif) disponible jusqu'au temps t , c'est à dire $X^t = (X_s)_{0 \leq s \leq t}$ et $\xi_t = \psi(X_t)$ avec ξ notre stratégie de trading.

Il peut clairement avoir une infinité de choix pour ξ , ainsi pour reduire la complexité du problème consistant à trouver une stratégie optimale, il est courant de se restreindre à une certaine classe de stratégies .

Nous nous restreignons ici à des stratégies de la forme $\xi_t = \langle \ell, \mathbb{X}_{0,t} \rangle$ où $\mathbb{X}_{0,t} \in T(\mathbb{R})$ est la signature du processus X^t et $\ell \in T(\mathbb{R}_*)$ est une fonctionnelle linéaire qui accepte des objets de type signature comme $\mathbb{X}_{0,t}$ comme input. En fait , on exprime juste ξ_t comme une combinaison linéaire des termes de la signature du processus considéré.

En pratique , la taille de la signature est infinie, donc on utilise la N^{ime} troncature de la signature $\mathbb{X}_{0,t}^{\leq N}$ ($N \in \mathbb{N}$) ainsi que de la fonctionnelle $\ell \in T^N(\mathbb{R}_*)$, l'espace dual de la N^{ime} troncature de l'algèbre tensoriel de \mathbb{R} .

La question naturelle qui se pose alors est de savoir si la classe des stratégies qui découle des signatures peut être capable de répliquer les stratégies de trading existantes ou d'en inspirer de nouvelles. Cela est possible par le biais du théorème d'approximation universelle qui stipule que *les fonctions continues sur les chemins peuvent être arbitrairement bien approximées par des fonctionnelles linéaires sur la signature, à condition de se restreindre à un sous-ensemble compact de chemins continus.*

Théorème 1.1. (*Approximation universel, voir Theorem 3.1 dans [4]*).

Soit $K \subset C^1([0,T], \mathbb{R}^d)$, un sous-ensemble compact de chemins. $\forall \phi \in C(K), \forall \epsilon > 0, \exists$ une fonctionnelle linéaire $\ell \in T((\mathbb{R}^d)_*)$ telle que :

$$\sup_{X \in K} \|\phi(X) - \langle \ell, \mathbb{X}_{0,T} \rangle\| < \epsilon$$

où le choix d'une topologie candidate appropriée est discuté dans le document [5].

Après avoir motivé l'utilisation du concept de signature en *trading* quantitatif, nous allons dès à présent présenter les fondements mathématiques inhérents à cet objet après avoir introduit la notion d'algèbre tensorielle, qui est l'espace dans lequel vit l'objet signature.

1.2 Préliminaires d'algèbre tensorielle

Définition 1.1. (Puissance tensorielle).

Soit E un espace vectoriel et k un entier positif. Nous définissons la k -ième puissance tensorielle de E comme :

$$E^{\otimes k} = \underbrace{E \otimes \dots \otimes E}_{k \text{ fois}}$$

où \otimes est le produit tensoriel entre deux espaces vectoriels. Dans la littérature, la convention est faite que $E^{\otimes 0} = K$ où K est le corps sur lequel E est défini.

Il découle que, si E est un espace vectoriel de dimension finie d avec une base $\mathcal{B} = (e_1, \dots, e_d)$, alors $E^{\otimes k}$ est l'espace vectoriel avec comme base $\mathcal{B}_k = (e_{i_1, \dots, i_k})_{1 \leq i_1, \dots, i_k \leq d}$, tel que $\dim(E^{\otimes k}) = \dim(E)^k$.

Tout élément de $E^{\otimes k}$, est appelé tenseur, et son rang est k . Naturellement, les tenseurs avec dimension 0, 1, 2 sont appelés respectivement scalaires, vecteurs, matrices.

Définition 1.2. (Algèbre tensorielle).

Soit E un espace vectoriel. L'algèbre tensorielle de E n'est rien d'autre que l'algèbre de tenseurs définis sur E , tel que le produit entre deux éléments est le produit tensoriel \otimes . Ainsi, si nous désignons par $T(E)$ l'algèbre tensorielle de E , alors on peut écrire :

$$T(E) = \bigoplus_{k=0}^{\infty} E^{\otimes k}$$

En fait, si on se limite à E , un espace vectoriel de dimension finie d avec une base $B = (e_1, \dots, e_d)$, alors un élément $z \in T(E)$ peut être écrit $z = (z^0, z^1, \dots, z^k, \dots)$, où $z^k = (\alpha_{i_1, \dots, i_k})_{1 \leq i_1, \dots, i_k \leq d} \in E^{\otimes k}$ est un tenseur de rang k . z peut s'apparenter à une série de tenseurs de taille croissante.

De même, pour $x, y \in T(E)$, on a :

$$\begin{aligned} x \otimes y &= \sum_{k=0}^{\infty} \left(\sum_{1 \leq i_1, \dots, i_k \leq d} x^{i_1, \dots, i_k} e_{i_1, \dots, i_k} \right) \otimes \left(\sum_{l=0}^{\infty} \left(\sum_{1 \leq j_1, \dots, j_l \leq d} y^{j_1, \dots, j_l} e_{j_1, \dots, j_l} \right) \right) \\ &= \sum_{m=0}^{\infty} \sum_{k+l=m} \sum_{1 \leq i_1, \dots, i_k, j_1, \dots, j_l \leq d} x^{i_1, \dots, i_k} y^{j_1, \dots, j_l} (e_{i_1, \dots, i_k} \otimes e_{j_1, \dots, j_l}) \\ &= \sum_{m=0}^{\infty} \sum_{k+l=m} \sum_{1 \leq i_1, \dots, i_k, j_1, \dots, j_l \leq d} x^{i_1, \dots, i_k} y^{j_1, \dots, j_l} (e_{i_1, \dots, i_k, j_1, \dots, j_l}) \end{aligned}$$

Ceci n'est rien d'autre qu'un produit de Cauchy entre séries réelles infinies.

Définition 1.3. (Algèbre tensorielle tronquée d'ordre N).

Soit E un espace vectoriel et N un entier naturel. L'algèbre tensorielle tronquée N de E est le sous-ensemble de l'algèbre tensorielle de E contenant les éléments qui ont des tenseurs nuls pour tout rang $k > N$.

Plus explicitement, si on désigne par $T^{(N)}(E)$ l'algèbre tensorielle tronquée N de E , alors, on a :

$$T^{(N)}(E) = \{z = (z^0, z^1, \dots, z^k, \dots) \in T(E), z^k = 0 \text{ pour tout } k > N\}$$

Comme précédemment, $T^{(N)}(E)$ contient des éléments de la forme $z = (z^0, z^1, \dots, z^N, 0, \dots)$, tel que si E est un espace vectoriel de dimension finie d avec une base $\mathcal{B} = (e_1, \dots, e_d)$, alors $\mathcal{B}^N = \bigcup_{k=0}^N \mathcal{B}_k$ est une base de $T^{(N)}(E)$, où $\mathcal{B}_k = (e_{i_1, \dots, i_k})_{1 \leq i_1, \dots, i_k \leq d}$. Il en découle que : $\dim(T^{(N)}(E)) = 1 + d + d^2 + \dots + d^N = \frac{d^{N+1}-1}{d-1}$.

De même que précédemment, pour $x, y \in T^{(N)}(E)$, $x \otimes y$ correspond au produit de Cauchy entre séries réelles finies suivant :

$$\begin{aligned} x \otimes y &= \sum_{k=0}^N \left(\sum_{1 \leq i_1, \dots, i_k \leq d} x^{i_1, \dots, i_k} e_{i_1, \dots, i_k} \right) \otimes \left(\sum_{l=0}^N \left(\sum_{1 \leq j_1, \dots, j_l \leq d} y^{j_1, \dots, j_l} e_{j_1, \dots, j_l} \right) \right) \\ &= \sum_{m=0}^{2N} \sum_{k+l=m} \sum_{1 \leq i_1, \dots, i_k, j_1, \dots, j_l \leq d} x^{i_1, \dots, i_k} y^{j_1, \dots, j_l} (e_{i_1, \dots, i_k} \otimes e_{j_1, \dots, j_l}) \\ &= \sum_{m=0}^{2N} \sum_{k+l=m} \sum_{1 \leq i_1, \dots, i_k, j_1, \dots, j_l \leq d} x^{i_1, \dots, i_k} y^{j_1, \dots, j_l} (e_{i_1, \dots, i_k, j_1, \dots, j_l}) \end{aligned}$$

On peut aussi avoir la même somme, mais avec $m \in \{0, \dots, N\}$ si nous considérons la troncature d'ordre N de $T^{(2N)}(E)$ i.e, si on se limite aux tenseurs de rang inférieur ou égal à N .

Définition 1.4. (Indexation tensorielle comme dans le papier).

Soit E un espace vectoriel de dimension finie avec une base $\mathcal{B} = (e_1, \dots, e_d)$, et soit $x \in T(E)$. Alors $x^k = (a_{i_1, \dots, i_k})_{1 \leq i_1, \dots, i_k \leq d} \in E^{\otimes k}$ est un tenseur de rang k . Nous avons besoin de multi-indices pour indexer x^k et nous utiliserons un ensemble de mots w pour représenter ces multi-indices.

Par exemple, $x_{\mathbf{211}}^3 = a_{211}$ et $x_{\mathbf{0211}}^5 = a_{0211}$. En fait, nous pouvons utiliser le même indexage pour x lui-même puisque la longueur du mot w indiquera à quel tenseur nous souhaitons accéder. Par exemple, si $w = \mathbf{02}$ nous accédons au tenseur x^2 de rang 2, et à l'intérieur de ce tenseur, nous prenons la coordonnée 0 suivant la première dimension et la coordonnée 2 suivant la deuxième dimension.

Dans la suite, on notera W_i^j (resp. $W_i^{\leq j}$) l'ensemble des mots de longueur L (resp. longueur $\leq L$) dans l'alphabet $\mathcal{A}_i = \{\emptyset, 0, 1, \dots, i-1\}$. Par exemple,

$$W_3^{\leq 2} = \{\emptyset, 0, 1, 2, 00, 01, 02, 10, 11, 12, 20, 21, 22\}$$

Le nombre d'éléments dans W_{ij} est noté $|W_i^j| = \sum_{k=0}^j i^k$

Définition 1.5. (Fonctionnelles linéaires sur l'algèbre tensorielle).

Notez qu'il existe un appariement naturel entre l'algèbre tensorielle étendue $T(\mathbb{R}^d)$ et son espace dual $T((\mathbb{R}^d)^*)$, que nous notons $\langle \cdot, \cdot \rangle : T((\mathbb{R}^d)^*) \times T(\mathbb{R}^d) \rightarrow \mathbb{R}$ et qui est donné par

$$\langle \ell, \mathbb{X} \rangle = \sum_{w \in \mathcal{A}_Z} \ell_w \mathbb{X}^w$$

pour $\ell \in T((\mathbb{R}^d)^*)$, $\mathbb{X} \in T(\mathbb{R}^d)$,

1.3 Notion de signature

Définition 1.6. Définition.

Soit $X \in C([0, T], \mathbb{R}^d)$ un chemin lisse. Alors la transformation en signature de X entre l'intervalle de temps s et t où $0 \leq s \leq t \leq T$ est donnée par :

$$\mathbb{X}_{s,t} = (1, \mathbb{X}_{s,t}^1, \dots, \mathbb{X}_{s,t}^k, \dots) \in T(\mathbb{R}^d)$$

où

$$\mathbb{X}_{s,t}^k = \int_{s < u_1 < \dots < u_k < t} dX_{u_1} \otimes \dots \otimes dX_{u_k} \in (\mathbb{R}^d)^{\otimes k}$$

c'est-à-dire

$$\mathbb{X}_{s,t}^k = (\mathbb{X}_{s,t}^{i_1, \dots, i_k})_{1 \leq i_1, \dots, i_k \leq d}$$

où

$$\mathbb{X}_{s,t}^{i_1, \dots, i_k} = \int \dots \int_{s < u_1 < \dots < u_k < t} dX_{u_1}^{i_1} \dots dX_{u_k}^{i_k} \in \mathbb{R}$$

Intuitivement, $X_{s,t}^n$ est un tenseur de rang n contenant les scalaires $\mathbb{X}_{s,t}^{i_1, \dots, i_n}$ pour tous les n -uplets (i_1, \dots, i_n) dans $\{1, \dots, d\}^n$.

Pour ne pas alourdir les notations, la signature $\mathbb{X}_{s,t}$ peut aussi être écrite $S(X)_{s,t}$. De plus, quand nous faisons référence à la signature de X sans mentionner explicitement s et t , nous nous référons à la signature sur $[0, T]$. Dans ce cas, nous pouvons simplement écrire X ou $S(X)$ sans spécifier l'intervalle de temps sur lequel nous calculons la signature.

Un autre attribut de la signature, c'est qu'elle peut être définie de manière récursive, ce qui peut être très utile pour démontrer certaines de ses propriétés.

Définition 1.7. (Signature tronquée).

Soit $X \in C([0, T], \mathbb{R}^d)$ un chemin lisse. Alors sa signature N -tronquée, notée $\mathbb{X}^{\leq N}$ ou $S(X)^{\leq N}$, est la troncature de sa signature jusqu'au tenseur de rang N , c'est-à-dire :

$$\mathbb{X}^{\leq N} = (1, \mathbb{X}^1, \dots, \mathbb{X}^N) \in T^{(N)}(\mathbb{R}^d)$$

1.3.1 Quelques intuitions

Afin de mieux comprendre et d'avoir des intuitions relatives à ce concept, nous pouvons étudier son comportement dans de petites dimensions et avec des chemins linéaires afin de mieux la comprendre.

Cas $d = 1$

Considérons d'abord le cas d'un chemin unidimensionnel. Dans ce cas, On note que $\mathbb{R}^{\otimes k} \cong \mathbb{R}$ donc que $T(\mathbb{R}) \cong \mathbb{R}^N$ c'est-à-dire, la signature de $X \in C([0, T], \mathbb{R})$ est simplement une série numérique indexée par N . Les premiers termes sont faciles à calculer :

$$\mathbb{X}^1 = \int_0^T dX_u = X_T - X_0$$

$$\mathbb{X}^2 = \iint_{0 < u < v < T} dX_u dX_v = \int_0^T dX_u \int_0^u dX_v$$

$$\mathbb{X}^2 = \int_0^T dX_u (X_u - X_0) = X_T(X_T - X_0) - \frac{(X_T - X_0)^2}{2} = \frac{(X_T - X_0)^2}{2}$$

Ainsi, on voit que le terme \mathbb{X}^1 peut être interprété comme la variation totale de X tandis que le second terme \mathbb{X}^2 peut être considéré comme la variation quadratique de X .

Il existe une certaine similitude avec les moments d'une variation aléatoire, mais la comparaison ne s'étend pas plus loin et les termes d'ordre supérieur de la signature sont plus difficiles à interpréter. Ce qui importe crucialement, c'est que ces termes d'ordre supérieur encodent des propriétés géométriques complexes et non linéaires du chemin. (par conséquent, ils sont de bons candidats pour capter des signaux sophistiqués cachés dans un processus stochastique et donc sont d'un vrai intérêt dans des problématiques de conceptions de stratégies basées sur la génération de signaux)

La même interprétation s'applique au cas où $d > 1$, comme montré ci-dessous.

Cas $d > 1$

De même, lorsque $d > 1$, nous pouvons interpréter le vecteur \mathbb{X}^1 comme la variation totale de X dans chaque dimension et la matrice \mathbb{X}^2 peut être considérée comme la variation quadratique de X entre chaque dimension. En effet, nous avons :

$$\mathbb{X}^1 = \int_0^T dX_u = X_T - X_0 = (X_T^i - X_0^i)_{1 \leq i \leq d} \in \mathbb{R}^d$$

$$\mathbb{X}^2 = \iint_{0 < u < v < T} dX_u \otimes dX_v = \left(\iint_{0 < u < v < T} dX_u^i dX_v^j \right)_{1 \leq i, j \leq d} \in \mathcal{M}_d(\mathbb{R})$$

Cas d'un chemin linéaire

Considérons maintenant un chemin linéaire $(a+bt)$ où $a, b \in \mathbb{R}^d, 0 \leq t \leq T$. Dans ce cas, on a :

$$\mathbb{X}_{s,t}^{i_1, \dots, i_k} = b_{i_1} \cdot \dots \cdot b_{i_k} \frac{(t-s)^k}{k!}$$

En particulier, le $k!$ garantit que les termes de la signature décroissent à mesure que le rang tensoriel k augmente. En d'autres termes, la signature converge.

Remarquez que si $d = 1$, alors

$$\mathbb{X}_{s,t} = \left(1, b \frac{(t-s)}{1!}, -b^2 \frac{(t-s)^2}{2!}, b^3 \frac{(t-s)^3}{3!}, \dots, (-1)^k b^k \frac{(t-s)^k}{k!} \right)$$

ce qui correspond à l'expansion en série de $\exp(b(t-s))$.

1.3.2 Propriétés

Invariance par translation

Puisque la signature dépend seulement des intégrales itérées des différentielles de X le long de chaque dimension, elle est invariante par translation, c'est-à-dire pour tout chemin lisse $X \in C([0, T], \mathbb{R}^d)$ et pour tout $a \in \mathbb{R}^d$, $S(X) = S(X + a)$.

Invariance par reparamétrisation du temps

Étant donné une bijection croissante définie : $\varphi : [T_1, T_2] \rightarrow [0, T]$, nous pouvons définir le chemin reparamétrisé par le temps $X^\varphi = X \circ \varphi : [T_1, T_2] \rightarrow \mathbb{R}^d$. Alors $S(X^\varphi) = S(X)$.

Cela vient du fait que pour tout chemin $X, Y \in C([0, T], \mathbb{R})$, $\int_{T_1}^{T_2} X_t^\varphi dY_t^\varphi = \int_0^T X_t dY_t$, combiné avec la définition récursive de la signature.

Identité de Chen Considérons deux chemins $X^1 : [a, b] \rightarrow \mathbb{R}^d$ et $X^2 : [b, c] \rightarrow \mathbb{R}^d$ où $a < b < c$. Nous définissons leur concaténation $X = X^1 * X^2$ par $X_t = X_t^1$ pour tout $t \in [a, b]$ et $X_t = X_b^1 + (X_t^2 - X_b^2)$ pour tout $t \in [b, c]$. Alors l'identité de Chen nous dit que

$$S(X) = S(X^1) \otimes S(X^2)$$

En particulier, l'identité de Chen nous permet de calculer la signature de tout chemin discret, et puisque tout chemin stocké sur un ordinateur doit être discret, cela nous permet effectivement de calculer toute signature. En effet, un chemin discret a des instants $(t_i)_{0 \leq i < p}$ et chaque sous-chemin $X^i = (X_t)_{t_i \leq t < t_{i+1}}$ est linéaire, nous pouvons donc facilement calculer sa signature, et l'identité de Chen donne $S(X) = S(X^0) \otimes S(X^1) \otimes \dots \otimes S(X^{p-1})$ tel que nous puissions calculer la signature finale comme un produit tensoriel de plusieurs signatures simples.

Complexité temporelle du calcul de la signature

Nous pouvons estimer la complexité temporelle du calcul de la signature en utilisant la méthode de Chen pour le calcul de la signature. Notez que les bibliothèques de signature utilisent également la méthode de Chen mais avec de légères modifications pour accélérer le calcul, y compris la parallélisation. Jetons un coup d'œil à la complexité temporelle du calcul de $X_{0,T}^{\leq N}$:

- Pour un chemin linéaire $X : t \in [0, T] \mapsto a + bt \in \mathbb{R}^d$: la complexité est $O(d^N)$.
Preuve. En utilisant l'équation (2), nous avons que $\mathbb{X}_{t_0, T}^{i_1, \dots, i_k} = b_{i_1} \dots b_{i_k} \frac{(T-t_0)^k}{k!}$, ce qui permet de calculer \mathbb{X}^k à partir de \mathbb{X}^{k-1} car chacun de ses d^k termes est calculé en $O(1)$. Par conséquent, $\mathbb{X}^{\leq N}$ est calculé récursivement en $O(1 + d + d^2 + \dots + d^N) = O\left(\frac{d^{N+1}-1}{d-1}\right) = O(d^N)$.
- Pour un chemin discret (c'est-à-dire linéaire par morceaux) $X : t \in \{t_i\}_{0 \leq i \leq p} \mapsto X_t \in \mathbb{R}^d$: la complexité est $O(\log_2(p)d^{2N})$.
Preuve. En effet, en utilisant l'équation (3) nous avons $S(X)^{\leq N} = S(X^0)^{\leq N} \otimes S(X^1)^{\leq N} \otimes \dots \otimes S(X^{p-1})^{\leq N}$ où $S(X^i)^{\leq N}$ est la signature N -tronquée de X^i qui est $(X_{t_i}, X_{t_{i+1}})$. En utilisant la multiplication rapide, cela revient à calculer $\log_2(p)$ produits de la forme $X \otimes Y$ où $X, Y \in T^{(N)}(\mathbb{R}^d)$. Ces produits sont calculés en $O(1 + d^2 + d^4 + \dots + d^{2N}) = O\left(\frac{d^{2N+1}-1}{d^2-1}\right)$ selon l'équation (1). Ainsi, calculer $X^{\leq N}$ est en $O(\log_2(p)d^{2N})$.

Pour conclure, la méthode de Chen nous permet de calculer la signature des chemins discrets en $O(\log_2(p)d^{2N})$ où d est la dimension du chemin, p est le nombre d'étapes de temps, et N est la profondeur de la signature. Nous remarquons que la dépendance en p est satisfaisante, comme l'est la dépendance en d (pas de malédiction de dimensionnalité). Cependant, la complexité est exponentielle en N . Cette dernière remarque est d'une grande

importance comme nous le verrons dans la section suivante : la complexité exponentielle en N crée un goulet d'étranglement computationnel lorsqu'on tente notre algorithme de Signature Trading à des niveaux de profondeur élevés.

À quel ordre de troncature N peut-on s'arrêter ?

Théorème 1.2. *Soit X un chemin de dimension d à variation bornée. Alors, étant donné $1 \leq i_1, \dots, i_n \leq d$, on a :*

$$\left\| \int_0^{u_1} \dots \int_0^{u_n} dX_{u_1}^{i_1} \dots dX_{u_n}^{i_n} \right\| \leq \frac{\|X\|_1^n}{n!}$$

avec

$$\|X\|_1 = \sup_{\{t_j\} \subseteq [0, T]} \sum_i |X_{t_{i+1}} - X_{t_i}|,$$

où $\|X\|_1$ est la norme de la variation totale de X sur $[0, T]$.

Ce théorème nous garantit théoriquement que si nous ne considérons que les premiers termes de la signature, nous ne perdons pas beaucoup d'informations puisque les termes (d'ordre supérieur) abandonnés ont une décroissance factorielle. Par conséquent, la signature tronquée d'un chemin décrit bien le chemin, et nous pouvons donc l'utiliser comme une caractéristique du chemin.

1.3.3 Transformation de trajectoires

Avant de calculer la signature d'un chemin, on peut appliquer certaines opérations de transformation. Des transformations judicieuses peuvent en effet rendre la signature subséquente plus éclairante. Nous présentons deux de ces transformations que nous utilisons dans notre contexte : la transformation lead-lag et l'augmentation temporelle.

Augmentation temporelle

Étant donné un processus $X \in (\mathbb{R}^d)^{[0, T]}$, nous pouvons définir le processus augmenté temporellement $\hat{X} \in (\mathbb{R} \times \mathbb{R}^d)^{[0, T]}$ avec $\hat{X}_t = (t, X_t), \forall t \in [0, T]$. Par exemple, pour un chemin discret $X = (2, 3, 5, 7)$, le processus augmenté temporellement correspondant correspond à :

$$\hat{X} = \left(\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix} \right)$$

Intérêt

Une caractéristique intéressante des chemins augmentés temporellement est que nous avons une assurance théorique de l'unicité de leur signature. Voir [6] pour de plus amples explications.

Transformation Lead-Lag

Il ya plusieurs types de transformations lead lag. Nous discutons ici, une méthode "naïve" et une autre dite de Hoff :

Transformation intuitive

Étant donné un chemin discret $X : t \in \{t_i\}_{1 \leq i \leq p} \mapsto X_t \in \mathbb{R}^d$, nous pouvons définir sa transformation lead-lag $(X_{t_i}^{LL})_{1 \leq i \leq 2p-1}$ où

$$X_{t_i}^{LL} = \begin{cases} (X_{t_i}, X_{t_{i+1}}) & \text{si } i \text{ est pair,} \\ (X_{t_{i+1}}, X_{t_i}) & \text{si } i \text{ est impair.} \end{cases}$$

Ainsi, $X^{LL} = (X^{lead}, X^{lag}) \in \mathbb{R}^d \times \mathbb{R}^d$ contient à la fois un signal principal et un signal de suivi pour chaque instant. En particulier, notez que si X est un chemin discret en dimension d avec p pas, alors X^{LL} est un chemin discret en dimension $2d$ avec $2p - 1$ pas.

Dans le cadre de notre étude nous allons prioriser la variante lead-lag introduite par [7], qui se base sur les travaux de Ben Hoff [7].

Définition 1.8. Processus Lead Lag de Hoff

Soit $\hat{X} : [0, T] \rightarrow \mathbb{R}^{d+1}$

le processus augmenté en temps continu de X , échantillonné de manière discrète aux instants $t = t_0, \dots, t_{2N}$. Le chemin transformé lead-lag de Hoff est défini comme l'interpolation linéaire par morceaux $\hat{X}^{LL} : [0, T] \rightarrow \mathbb{R}^{2(d+1)}$ tel que

$$\hat{X}_{t_i}^{LL} = (\hat{X}_{t_i}^{lead}, \hat{X}_{t_i}^{lag})_{i=1}^{2N}$$

où

$$\hat{X}_t^{lead} = \begin{cases} \hat{X}_{t_{k+1}}, & \text{si } t \in [2k, 2k+1] \\ \hat{X}_{t_{k+1}} + 2(t - (2k+1))(\hat{X}_{t_{k+2}} - \hat{X}_{t_{k+1}}), & \text{si } t \in [2k+1, 2k+\frac{3}{2}] \\ \hat{X}_{t_{k+2}}, & \text{si } t \in [2k+\frac{3}{2}, 2k+2], \end{cases}$$

$$\hat{X}_t^{lag} = \begin{cases} \hat{X}_{t_i}, & \text{si } t \in [2k, 2k+\frac{3}{2}] \\ \hat{X}_{t_{k+1}} + 2(t - (2k+\frac{3}{2}))(\hat{X}_{t_{k+1}} - \hat{X}_{t_k}), & \text{si } t \in [2k+\frac{3}{2}, 2k+2]. \end{cases}$$

Ci-dessous une illustration de la transformation lead-lag selon Hoff, du cours de clôture du S&P 500 entre le 01-03-2020 et le 30-04-2020 :

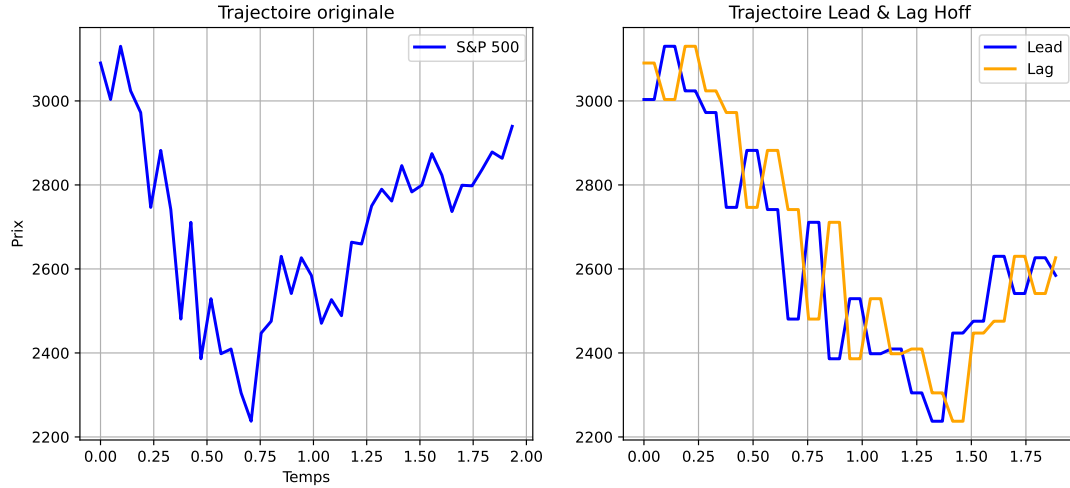


FIGURE 1 – Le S&P 500 et sa transformation lead-lag

Intérêt de la transformation lead-lag (Hoff)

En ayant accès aux composantes lead et lag par la transformation effectuée, nous avons accès à la *variation* du chemin considéré. Cela peut être intéressant pour capturer certains signaux (variation quadratique par exemple).

En machine learning, le *feature engineering* se fondant sur la signature du signal préalablement transformée sous forme lead-lag améliore considérablement les performances par rapport à la signature du chemin vanille.

Dans notre contexte, nous choisirons dans la suite, la transformée de Hoff en raison de ses propriétés fondamentales qui facilitent le calcul de notre PnL dont celle démontrée dans [7] qui stipule que l'intégrale d'Itô d'une fonction d'un processus X par rapport à elle-même, peut être récupérée via les composantes de la transformation lead-lag de Hoff (détaillé dans la section ??).

1.4 Quelques domaines d'utilisation de la signature

Plusieurs usages de ce concept sont possibles, notamment la reconnaissance de caractère.

Étant donné que la signature est un objet qui encode les propriétés géométriques d'une trajectoire, c'est donc un candidat naturel pour le *feature engineering* et les tâches de classification.

À ce propos, des tests simplistes de classification ont été effectués dans un *notebook* dans le cadre de ce projet.

L'expérience consiste à générer des processus aléatoires qui suivent des modèles ARMA, à leur attribuer des étiquettes (-1, 1) puis à utiliser la signature des chemins comme *feature* avec une régression logistique puis une comparaison est effectuée avec un réseau de neurones à une couche cachée. on utilise 2000 échantillons de taille 1000 pour calibrer nos modèles. Les expériences sont réalisées sur un ordinateur Ryzen 5, 16 gb ram.

Ci-dessous un résumé des résultats (intervalle de confiance d'amplitude de 5%) obtenus :

Algorithme	Temps d'exécution (s)	depth	Performance (%)
reg.log. raw data	6s	-	50 %
reg.log. signature data naïve	1s	2	100 %
reg.log. signature data Hoff	2s	2	100 %
réseau de neurones raw data	37s	-	53 %
réseau de neurones signature Hoff	42s	2	100 %

TABLE 1 – Comparaison des performances des algorithmes pour un modèle ARMA (2,2)

De plus amples détails sont fournis dans le notebook qui permet aussi de faire d'autres tests pour voir la pertinence de la signature dans un problème assez classique et basique de classification. Ce qu'il est essentiel de retenir c'est que de façon claire, en utilisant la transformation de Hoff on a des performances au moins équivalentes (généralement meilleures) à ceux qu'on a en n'utilisant que les données brutes.

De plus, en utilisant les signatures comme *feature*, on a un gain de temps considérable pour des performances acceptables en comparaison avec ceux qu'on a avec des réseaux de neurones simples.

2 Signature trading

Nous entrons dans le vif de notre sujet relatif à la mise en place de stratégie de trading quantitatif avec le concept de signature.

2.1 Cadre de modélisation

Dans cette section, nous introduisons les ingrédients nécessaires à la détermination des stratégies de trading optimales.

L'idée centrale est fondée sur le théorème 1.1 qui nous assure que toute stratégie de trading peut être approximée par une fonctionnelle linéaire de signature de trajectoires.

On se place dans le cas général des marchés où on observe les trajectoires de prix et des facteurs qui enrichissent la filtration naturelle (informations) associé au processus de prix. On parle généralement de facteur exogène $(f_t)_{t \in [0, T]}$, $f_t = (f_t^1, \dots, f_t^N)$. Un *facteur de marché* peut être tout *feature* qu'on ne trade pas, classifié comme un processus stochastique $(f_t)_{t \in [0, T]}$ qui peut être utilisé afin d'enrichir l'état du marché et augmenter la prédictibilité. En pratique, les facteurs s'appréhendent comme des signaux directeurs du processus sous-jacent.

Définition 2.1. (Processus *Market factor*). Soit $X = (X_t)_{t \in [0, T]}$ un processus d'actifs *tradable* de dimension d et $\{f_t^i\}_{i=1}^N$ des signaux de trading exogènes (qu'on ne *trade* pas). Nous définissons alors le processus time-augmented market factor comme le processus $(1 + d + N)$ -dimensionnel : $\hat{Z}_t := (t, X_t, f_t)$

qui induit l'espace de probabilité filtré naturel $(\Omega, \mathcal{F}^Z, (\mathcal{F}_t^Z)_{t \in [0, T]}, \mathbb{P})$, où la filtration de X satisfait $\mathcal{F}^X \subseteq \mathcal{F}^Z$ tel que X est *drivé* par $\{f_t^i\}_{i=1}^N$. Dans la suite, nous conservons les hypothèses inhérentes à \hat{Z} et on définit comme l'espace de toutes les trajectoires de facteurs de marché pour les actifs donnés X et les facteurs f :

$$\mathcal{Z}_{0,T}^f := \{\hat{Z}_t = (t, X_t, f_t) \mid X : [0, T] \rightarrow \mathbb{R}^d, f : [0, T] \rightarrow \mathbb{R}^N \text{ et } Z_0 = (0, 1, 1)\}$$

Remarque. Si on veut un univers sans signal exogène, on considérera simplement que les facteurs de marché sont le processus nul \emptyset .

Le processus *market factor* ayant été introduit, le nombre d'éléments dans l'espace des mots W_{ij} , introduit en section 1.4 devient $|W_i^j| = |W_{d+N+1}^M| \sum_{k=0}^M (d + N + 1)^k$ avec en pratique d tel que $X = (X^1, \dots, X^d)$, N étant le même que dans $\{f_t^i\}_{i=1}^N$ et pour finir, M étant l'ordre de troncature.

Notre objectif, c'est de maximiser notre *PnL* espéré (de manière dynamique) sous certaines contraintes. Sous le prisme du *signature trading*, introduisons alors la définition du *PnL* :

Définition 2.2. (PnL de la Stratégie de trading).

Soit $\hat{Z} = (t, X_t, f_t)$ un *market factor process*, X un processus sous-jacent *tradable* et f le processus qui induit notre signal. Si on reprend notre ξ tel que $\xi_t^m = \langle \ell_m, \hat{Z}_{0,t} \rangle$, pour chaque actif $m = 1, \dots, d$, alors nous définissons le PnL de la stratégie de signature entre le temps 0 et le temps T comme :

$$PnL = \sum_{m=1}^d \int_0^T \langle \ell_m, \hat{Z}_{0,t} \rangle dX_t^m$$

où l'intégrale s'entend au sens d'Itô.

Il est crucial de distinguer l'intégrale d'Itô utilisée dans cette définition par rapport à l'intégrale de Stratonovich (voir littérature). En fait, il n'est pas aisée de manipuler l'intégrale d'Itô pour optimiser le PnL sous certaines contraintes (dans notre contexte). Une bonne idée est donc de se servir de la transformation *lead-lag* de Hoff afin de faire intervenir l'intégrale de Stratonovich

Pour ne pas alourdir le présent rapport, certains détails théoriques ne sont pas fournis mais au besoin le lecteur pourra se référer à notre papier référence [3].

$$\int_0^T \langle \ell_m, \hat{Z}_{0,t} \rangle dX_t^m \approx \int_0^T \langle \ell_m, \hat{Z}_{0,t} \rangle dX_t^{lead,m} = \langle \ell_m f(m), \hat{Z}_{0,T}^{LL} \rangle$$

où $f(m)$ correspond au “shift operator” introduit dans le papier. C'est juste une façon de capturer les éléments de $\hat{Z}_{0,T}^{LL}$ suivant le $(k+1)^{ime}$ tenseur qui stocke les signatures (vrai à mesure que $\sup_i |t_{i+1} - t_i|$ tend vers 0).

Rappelons que notre objectif est de maximiser le *PnL* espéré sous certaines contraintes comme le fait de limiter la variance. On a :

$$\mathbb{E}(PnL) = \langle \ell_m f(m), \mathbb{E}(\hat{Z}_{0,T}^{LL}) \rangle$$

$$Var(PnL) = \sum_{m=1}^d \sum_{n=1}^d \left(\langle \ell_m \mathbf{f}(m) \sqcup \ell_n \mathbf{f}(n), \mathbb{E}(\hat{Z}_T^{LL}) \rangle - \langle \ell_m \mathbf{f}(m), \mathbb{E}(\hat{Z}_T^{LL}) \rangle \langle \ell_n \mathbf{f}(n), \mathbb{E}(\hat{Z}_T^{LL}) \rangle \right)$$

Avec \sqcup , l'opérateur dit de mélange *shuffle*.

Définition 2.3. (shuffle product.)

Le produit shuffle , $\sqcup : W(\mathcal{A}_d) \times W(\mathcal{A}_d) \rightarrow W(\mathcal{A}_d)$ est défini de manière par :

$$\mathbf{ua} \sqcup \mathbf{vb} = (\mathbf{u} \sqcup \mathbf{vb})\mathbf{a} + (\mathbf{ua} \sqcup \mathbf{v})\mathbf{b}$$

$$\mathbf{w} \sqcup \emptyset = \emptyset \sqcup \mathbf{w} = \mathbf{w}$$

pour tous les *words* \mathbf{u}, \mathbf{v} et lettres $\mathbf{a}, \mathbf{b} \in W(\mathcal{A}_d)$.

Exemple Soient $\mathbf{w} = \mathbf{12}$, $\mathbf{v} = \mathbf{34}$, alors $\mathbf{w} \sqcup \mathbf{v}$ est donné par :

$$\mathbf{12} \sqcup \mathbf{34} = \mathbf{1234} + \mathbf{1324} + \mathbf{1342} + \mathbf{3124} + \mathbf{3142} + \mathbf{3412}$$

Sous le prisme mean-variance , la stratégie optimale est donnée par le théorème suivant (preuve dans le papier) :

Théorème 2.1. (Stratégie de Trading Optimale).

Soit $T \in \mathbb{N}$ le temps terminal. Soit X un processus stochastique tradable de dimension d et soit f un processus non tradable de dimension N .

Définissons $\hat{Z}_t := (t, X_t, f_t)$ comme le processus factoriel du marché. Définissons une stratégie de trading signature ξ_t^m à travers une fonctionnelle linéaire sur la signature du processus factoriel du marché, c'est-à-dire $\xi_t^m = \langle \ell_m, \hat{Z}_{0,t} \rangle$. Alors, pour un niveau de troncature M donné, la stratégie de trading optimale moyenne-variance $\ell^* = (\ell_1^*, \dots, \ell_d^*)$, $\ell_m^* \in T^{\leq M}((\mathbb{R}^{N+d+1})^*)$, satisfait :

$$\ell_m^* := \arg \max_{\ell_m \in T^{\leq M}((\mathbb{R}^{N+d+1})^*)} \left\{ \sum_{m=1}^d \left\langle \ell_m \mathbf{f}(m), \mathbb{E} \left(\hat{Z}_{0,T}^{LL} \right) \right\rangle \right\}, \quad \forall m \in \{1, \dots, d\}$$

Et , on a :

$$\langle \ell_m^*, e_{\mathbf{w}} \rangle = \frac{(\Sigma^{sig})^{-1} \mu_{\mathbf{w}\mathbf{f}(m)}^{sig}}{2\lambda}, \quad m \in \{1, \dots, d\}, \mathbf{w} \in W_{N+d+1}^{\leq M}$$

où le paramètre de mise à l'échelle de la variance (variance-scaling) λ est donné par :

$$\lambda = \frac{1}{2\sqrt{\Delta}} \left(\sum_{m=1}^d \sum_{n=1}^d \sum_{\mathbf{w} \in W_{N+d+1}^M} \sum_{\mathbf{v} \in W_{N+d+1}^M} \left(((\Sigma^{sig})^{-1} \mu_{\mathbf{w}\mathbf{f}(m)}^{sig}) (\Sigma^{sig})^{-1} \mu_{\mathbf{v}\mathbf{f}(n)}^{sig} \Sigma_{\mathbf{w}\mathbf{f}(m), \mathbf{v}\mathbf{f}(n)}^{sig} \right) \right)^{\frac{1}{2}}.$$

Avec :

$$\mu^{sig} = \left(\mu_1^{sig}, \dots, \mu_d^{sig} \right)^T ; \mu_{\mathbf{w}\mathbf{f}(m)}^{sig} = \left\langle \mathbf{w}\mathbf{f}(m), \mathbb{E} \left(\hat{Z}_{0,T}^{LL} \right) \right\rangle$$

$$\Sigma_{\mathbf{w}\mathbf{f}(m), \mathbf{v}\mathbf{f}(n)}^{sig} = \langle \mathbf{w}\mathbf{f}(m) \sqcup \mathbf{v}\mathbf{f}(n), \mathbb{E}(\hat{Z}_T^{LL}) \rangle - \langle \mathbf{w}\mathbf{f}(m), \mathbb{E}(\hat{Z}_T^{LL}) \rangle \langle \mathbf{v}\mathbf{f}(n), \mathbb{E}(\hat{Z}_T^{LL}) \rangle$$

Intéressant, car il y a des formules fermées.

Petit exemple de compréhension

Considérons le cas où nous avons deux actifs $X = (X^1, X^2)$, un facteur f , de sorte que nous définissons le processus de facteur de marché 4-dimensionnel $Z := (t, X, f)$. Fixons le niveau de troncation à $M = 2$. Remarquons que l'Exemple 2.1 est de la même forme, et donc la signature tronquée de niveau 2 \hat{Z} a $|W_4^2| = 21$ termes, :

$$W_4^2 = \{\emptyset, 0, 1, 2, 3, 00, 01, 02, 03, 10, 11, 12, 13, 20, 21, 22, 23, 30, 31, 32, 33\}$$

Par conséquent, la stratégie de trading de signature linéaire optimale correspondra à deux fonctionnelles linéaires (une pour chaque actif) ℓ_1, ℓ_2 , chacune de longueur 21, définies comme suit :

$$\xi_t^1 = (\ell_1, \hat{Z}_{0,t}^{\leq 2}); \xi_t^2 = (\ell_2, \hat{Z}_{0,t}^{\leq 2}); \mu^{sig} := \mathbb{E} \left(\hat{Z}_{wf(m)}^{LL, \leq 3} \right).$$

Par conséquent, μ^{sig} sera un vecteur de longueur $d \cdot |W_4^M| + d + 1 = 2 \times 21 = 42$, contenant des éléments de la signature attendue lead-lag d'ordre 3.

Rappelons que l'opérateur de décalage $f(m)$ est défini pour chaque actif $m = 1, 2$ comme suit : $f(1)=5; f(2)=6$.

Ceci correspond aux cinquième et sixième dimensions du processus de lead-lag de Hoff. Ainsi, nous observons que I_1^{sig} contient les termes de signature de décalage attendus correspondant à l'index des mots :

$$I_1 = \{5, 05, 15, 25, 35, 005, 015, 025, 035, 105, 115, 125, 135, 205, 215, 225, 235, 305, 315, 325, 335\}$$

et I_2^{sig} contient les termes de signature de décalage attendus correspondant à l'index des mots :

$$I_2 = \{6, 06, 16, 26, 36, 006, 016, 026, 036, 106, 116, 126, 136, 206, 216, 226, 236, 306, 316, 326, 336\}$$

Intéressons nous maintenant à la matrice 42×42 définie élément par élément comme suit :

$$\Sigma_{wf(m), vf(n)}^{sig} = \left\langle wf(m) \sqcup vf(n), \mathbb{E} \left(\hat{Z}_{0,T}^{LL} \right) \right\rangle - \left(wf(m), \mathbb{E} \left[\tilde{Z}_{LL,\infty}^{(0,T)} \right] \right) \cdot \left(vf(n), \mathbb{E} \left[\tilde{Z}_{LL,\infty}^{(0,T)} \right] \right)$$

pour tout $w, v \in W^2$ et $m, n = 1, 2$.

Par exemple, observons un élément arbitraire de Σ^{sig} . Soit $w = 01, v = 23, m = 1, n = 2$. Alors $wf(m) = 015, vf(n) = 236$ et l'élément correspondant dans la matrice est donné par :

$$\Sigma_{015, 236}^{sig} = \left\langle 015 \sqcup 236, \mathbb{E} \left(\hat{Z}_{0,T}^{LL, \leq 6} \right) \right\rangle - \left\langle 015, \mathbb{E} \left(\hat{Z}_{0,T}^{LL, \leq 3} \right) \right\rangle \left\langle 236, \mathbb{E} \left(\hat{Z}_{0,T}^{LL, \leq 3} \right) \right\rangle$$

L'obtention de la solution optimale $\ell^* = (\ell_1^*, \ell_2^*)$, se fait via les formules fermées intro-

duites :

$$\ell^* = (\Sigma^{\text{sig}})^{-1} \mu^{\text{sig}}$$

Nous calculons ensuite la stratégie de trading, pour chaque instant t , comme suit :

$$\xi_t^1 = \langle \ell_1^*, \hat{Z}_{0,t}^{\leq 2} \rangle$$

$$\xi_t^2 = \langle \ell_2^*, \hat{Z}_{0,t}^{\leq 2} \rangle$$

Nous pouvons également calculer explicitement le PnL espéré et la variance du portefeuille dans ce cadre. Soit ℓ le vecteur de longueur 42, $\ell = (\theta_1^*, \theta_2^*)$, alors nous avons :

$$\mathbb{E}[PnL] = \ell^T \mu^{\text{sig}}$$

$$\text{Var}(PnL) = \ell^T \Sigma^{\text{sig}} \ell$$

Nous introduisons dès à présent l'algorithme pour mettre en œuvre tout ceci.

2.2 Algorithme sig-trader

La mise en œuvre se décline selon deux algorithmes : en premier lieu , il s'agit du *fitting* qui concerne la détermination des fonctionnelles encapsulées dans ℓ ensuite la partie trading qui consiste à ajuster les différentes positions .

Algorithm 1 Fitting The Optimal Signature Trading Strategy

Input: A finite set of M d -dimensional sample market paths $\{X_{t \in [0,T]}^i\}_{i=1}^M$
A finite set of M N -dimensional sample factor (signal) paths $\{f_{t \in [0,T]}^i\}_{i=1}^M$

Output: $\ell_m \in T((\mathbb{R}^{N+d+1})^*) \quad \forall m \in \{1, \dots, d\}$: The optimal trading strategy as a linear functional on the signature of the time-augmented path

Parameters: $N \geq 0$: Truncation level of the signature
 $\Delta \geq 0$: Maximum variance of PnL

- 1: Create the $(N+d+1)$ -dimensional market factor process $\hat{Z} = (t, X_t, f_t)$ to obtain $\{\hat{Z}_{t \in [0,T]}^i\}_{i=1}^M$.
- 2: Apply time-augmentation and lead-lag transformations to all market factor paths, to obtain $\{\hat{Z}_{t \in [0,T]}^{LL}\}_{i=1}^M$.
- 3: Compute the truncated signature (at order N) of each market factor path $\{\hat{Z}_{t \in [0,T]}^{LL, \leq N}\}_{i=1}^M$.
- 4: Calculate the empirical expected signature $\mathbb{E}(\hat{Z}^{LL, \leq N}) = \frac{1}{M} \sum_{i=1}^M \hat{Z}_{t \in [0,T]}^{LL, \leq N}$
- 5: Populate vector μ^{sig} as a subset of $\mathbb{E}(\hat{Z}^{LL, \leq N})$ terms.
- 6: Populate matrix $\Sigma_{\mathbf{w}, \mathbf{v}}^{\text{sig}}$ for each word $\mathbf{w}, \mathbf{v} \in \mathcal{W}_{N+d+1}$.
- 7: Solve the system of linear equations as to obtain $\ell_m \in T((\mathbb{R}^{N+d+1})^*) \quad \forall m \in \{1, \dots, d\}$ where

$$\langle \ell_m^*, e_{\mathbf{w}} \rangle = \frac{((\Sigma^{\text{sig}})^{-1} \mu^{\text{sig}})_{\mathbf{wf}(m)}}{2\lambda}, \quad m \in \{1, \dots, d\}, \mathbf{w} \in \mathcal{W}_{N+d+1}^M$$

- 8: **return** Linear functional $\ell_m \quad \forall m \in \{1, \dots, d\}$.
-

FIGURE 2 – Algorithme pour le fitting.

Une fois le *fitting* terminé, il suffit juste de calculer $\xi_t^m = \langle \ell_m^*, \hat{\mathbb{Z}}_{0,t}^{\leq N} \rangle$ via l'algorithme suivant :

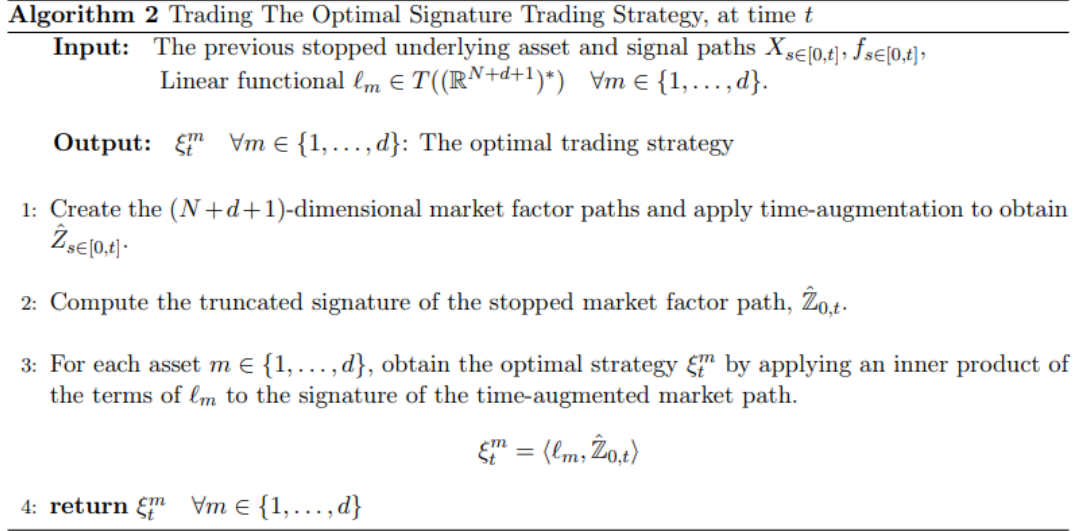


FIGURE 3 – Algorithme pour ajuster les positions au fil du temps.

Dans la suite, il est question d'évaluer cette méthode de construction de stratégies fondée sur la signature en voyant à quel point il est capable d'apprendre à capturer les signaux sous-jacents.

3 Quelques Applications

3.1 Cas Markowitz vs Signature trading

L'un des exemples les plus intéressants à analyser est sans doute le modèle de portefeuille optimal selon Markowitz. L'avantage, c'est qu'on connaît par formule fermée les poids optimaux à attribuer à chaque actif dans ce problème d'optimisation statique.

Sous ce prisme, si on a notre portefeuille d'actifs X tel que : $\log(X) \sim \mathcal{N}(\mu, \Sigma)$ alors les poids optimaux à affecter à chaque composante du portefeuille correspond à : $w_M = \frac{\Sigma^{-1}\mu}{2\lambda}$ avec λ qui correspond à l'inverse proportionnel de $\sqrt{\Delta}$.

Dans le cadre de notre signature trading, les fonctionnelles $\ell = (\ell_1, \dots, \ell_m)$ correspondent en théorie à ces poids. Avec : $\ell_m = \langle \ell_m, 1 \rangle = \langle \ell_m, \hat{\mathbb{Z}}_{0,t}^{\leq 0} \rangle = \xi_t^m$.

Mise en œuvre et résultats

L'algorithme *Signature trading* a été entraîné sur $M = 1000$ échantillons i.i.d. du même processus de prix logarithmique $\ln(X) \sim \mathcal{N}(\mu, \Sigma)$ avec $n = 10$ actifs (nos processus de prix) et sans signal exogène $f = \emptyset$.

μ a été générée de manière arbitraire selon une loi uniforme, et Σ a été simulée selon une matrice de *Hakenl* modifiée (symétrie, etc.), voir notebook pour les détails liés à l'implémentation. Les rendements ont été évalués sur une année de trading (soit 252 jours).

Les PnL des deux méthodes ainsi que leur ratio de sharpe ont été aussi comparé.
 Pour rappel , le ratio de Sharpe annuel S peut être exprimé comme suit :

$$S = \frac{\mu - r_f}{\sigma} \times \sqrt{N}$$

où :

- μ et σ correspondent à la moyenne et à l'écart-type des rendements ,
- r_f est le taux sans risque, ici supposé être 0 pour simplification,
- N est le nombre de jours de trading dans l'année, utilisé pour annualiser le ratio.

Ci-dessous des aperçus graphiques :

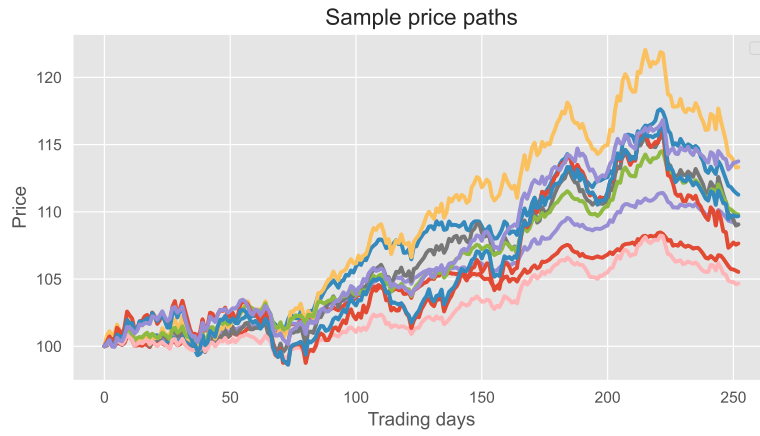


FIGURE 4 – Quelques trajectoires de notre processus de prix

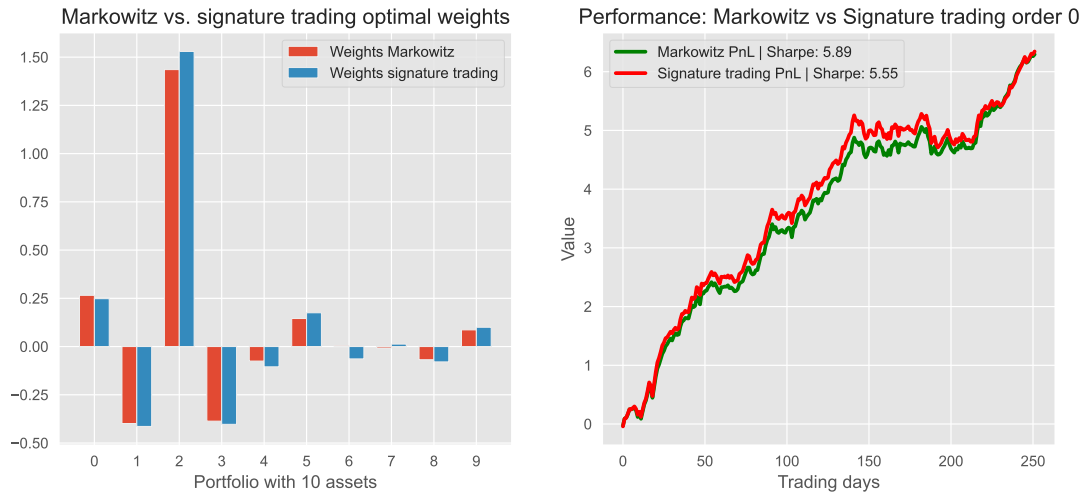


FIGURE 5 – Poids et performance du portefeuille optimal Markowitz vs. signature trading.

On constate que les poids obtenus avec notre *signature trading* et ceux obtenus selon Markowitz sont quasiment identiques et sans surprise, les deux *PnL* et ratio de Sharpe sont proches . Ainsi, avec une troncature d'ordre 0 , l'algorithme de trading basée sur la

signature des trajectoires ainsi que le *Signature Trader* d'ordre 0 est effectivement capable d'apprendre la stratégie statique optimale selon le critère moyenne-variance.

Nous allons à présent évaluer notre *signature trader* sur un signal de retour à la moyenne (*mean-reverting signal*), modèle très présent en trading algorithmique.

3.2 Stratégie *Mean-reversion*

Supposons que l'actif principal qu'on veut *trader* est une série temporelle unidimensionnelle $X = (X_t)_{t \in [0, T]}$ qui admet la propriété de retour à la moyenne vers la moyenne et qu'il n'y a pas de facteur exogène f ((dans notre cas un seul actif donc $d=1$)). Dans la littérature, X suit très souvent la dynamique d'Ornstein-Uhlenbeck définie par :

$$dX_t = \kappa(X_0 - X_t)dt + \sigma dW_t$$

le trader achète quand $X_t - X_0 < 0$ et vend quand $X_t - X_0 > 0$.

Comme on ne suppose pas de signal exogène, on a : $\hat{Z}_t = (t, X_t) \in \mathbb{R}^2$.

Nous testerons une troncature d'ordre 1 puisque ce modèle ne semble à priori pas très complexe. La stratégie se décline de la manière suivante :

$$\xi_t^m = \langle \ell, \hat{Z}_{0,t}^{\leq 1} \rangle = \ell_1^0 \cdot 1 + \ell_1^1 \cdot t + \ell_1^2 \cdot (X_t - X_0)$$

, en supposant que $\ell_1 = (\ell_1^0, \ell_1^1, \ell_1^2)$ puisque $\hat{Z}_{0,t}^{\leq 1} = \left(1, \begin{bmatrix} t \\ X_t - X_0 \end{bmatrix} \right)$.

Résultats

On suppose : $X_0 = 100$ (valeur autour de laquelle les trajectoires sont centrées) ; $\sigma^X = 0.07$ (volatilité annuel) ; $\kappa = 100$ (force de retour à la moyenne), période de trading : 45 jours ; M (taille de l'échantillon)=1000 ; fréquence des données = horaire.

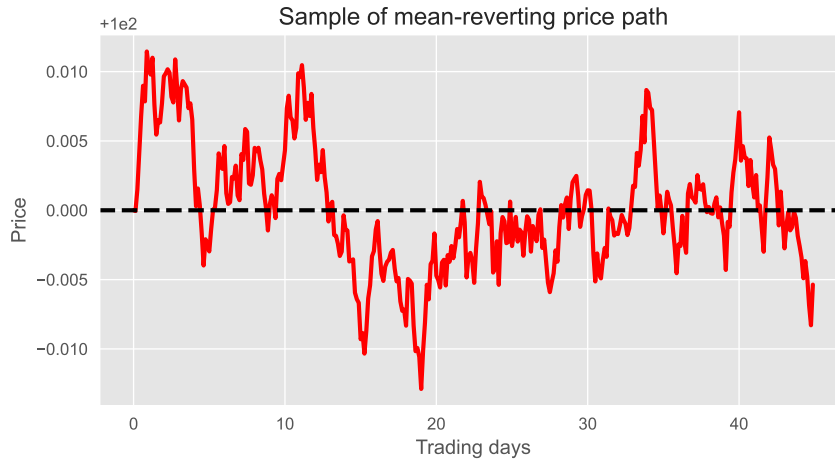


FIGURE 6 – Exemple de trajectoire *mean-reverting*

Ce qui est intéressant de voir à quel point notre signature trader est capable de capturer ce signal aussi bien sur les données de *fitting* (les données sur lesquelles on calibre les fonctionnelles ℓ_m) que sur d'autres données (*out-of-sample*) ayant les mêmes propriétés mais sur lesquelles les fonctionnelles n'ont pas été explicitement calibré (pour faire un parallèle avec des réseaux de neurones : on entraîne sur des données afin de trouver les poids optimaux et le biais ensuite ces poids optimaux sont utilisé pour la phase de test , sans besoin de le recalibrer le réseau). Ci-dessous des graphiques illustratifs des résultats :

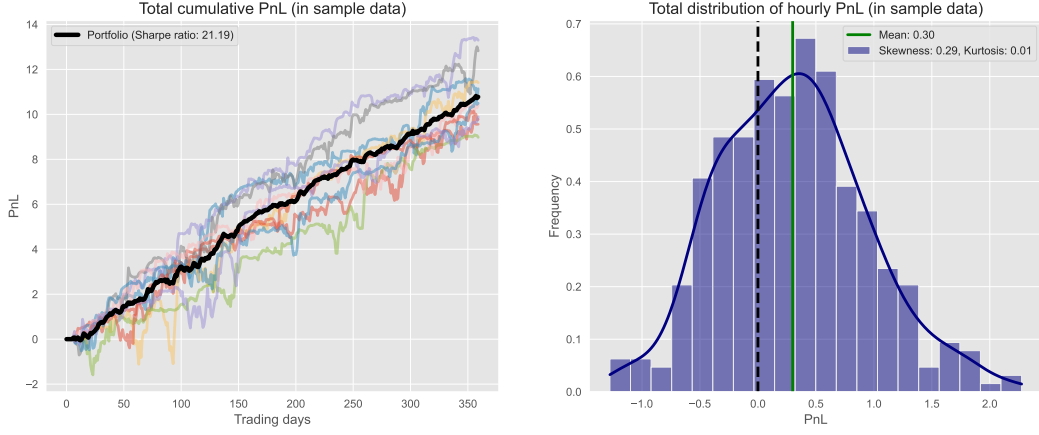


FIGURE 7 – Performance de notre *signature trading* sur les données *in-sample*.

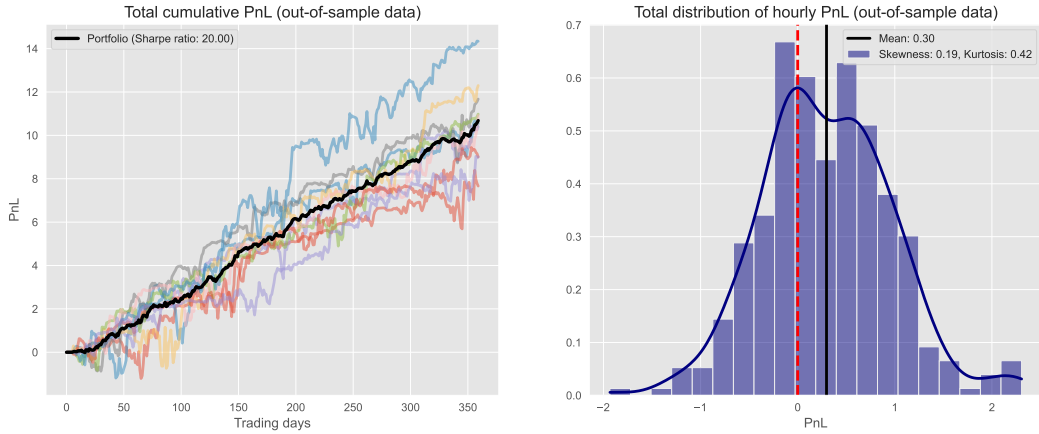


FIGURE 8 – Performance de notre *signature trading* sur les données *out-of-sample*.

On peut aussi observer les composantes de la fonctionnelle qu'on obtient et les auditer.

$$\ell_1 = \left(\underbrace{-32.54}_{\ell_1^0}, \left[\begin{array}{c} \underbrace{0.0709}_{\ell_1^1} \\ \underbrace{-1.57}_{\ell_1^2} \end{array} \right] \right)$$

Comme, le processus est de type *mean-reverting* et stationnaire , alors on s'attend à ce

que la composante associée au temps ℓ_1^1 tende vers 0 ce qui est bien le cas. Pour les autres composantes on ne saurait se prononcer.

En définitive , notre *signature trader* affiche des performances satisfaisantes à la fois sur les données en *in sample* et *out-of-sample*, ce qui indique qu'il se généralise bien sans *over-fitter*, et qu'il capture efficacement les signaux de trading de retour à la moyenne.

Intéressons nous à présent à une stratégie de type pair trading.

3.3 Stratégie de pairs trading

Tout d'abord, nous explorons le scénario dans lequel nous n'avons aucun signal de trading exogène pour enrichir notre stratégie de trading, mais nous avons seulement accès aux séries temporelles sous-jacentes du processus des actifs.

Le véritable avantage que les méthodes basées sur les données ont sur les cadres paramétriques classiques est qu'elles peuvent exploiter les inefficacités présentes dans les données des séries temporelles financières, sans spécifier les dynamiques explicites qu'elles tentent de capturer. Dans cet exemple simplifié, nous visons à isoler un aspect spécifique qui est exhibé par les données de séries temporelles et à mettre en évidence comment le Sig-Trader l'exploite.

Le pair trading est l'une des stratégies de trading actives les plus célèbres et originales déployées par les investisseurs, qui se concentre sur le trading du comportement conjoint entre deux actifs. L'idée est que, bien que les deux actifs aient leurs propres dynamiques, la différence (écart) entre les prix des deux actifs devrait avoir une certaine prévisibilité sur les futurs (co-)mouvements.

La littérature classique se concentre sur la modélisation de cette relation en utilisant un processus de retour à la moyenne tel qu'un processus d'Ornstein-Uhlenbeck. La stratégie devrait alors contenir un signal d'achat lorsque l'écart tombe en dessous d'un certain seuil et un signal de vente lorsque l'écart est au-dessus du seuil, en anticipant que cet écart devrait converger vers le seuil.

Dans notre contexte ici, on choisit deux actifs $X = (X_t)$ et $Y = (Y_t)$ tels que :

$$\begin{aligned} dX_t &= \sigma^X dW_t^X \\ dY_t &= \kappa(X_t - Y_t)dt + \sigma^Y dW_t^Y \end{aligned}$$

En résumé , on considère :

- Deux actifs ($d = 2$) ayant leurs dynamiques sous-jacentes individuelles mais on un comportement conjoint prévisible ;
- Le *spread* entre les deux actifs suit un processus de type *mean-reverting* ;
- Pas d'information exogène $f = \emptyset$ et $\hat{Z}_t = (t, X_t, Y_t)$ avec X_t notre premier actif et Y_t le second.

Comme discuté dans la partie du *mean-reverting* , le signature trader d'ordre 1 suffit à capturer un signal de type mean-reverting. Le raisonnement ici sera donc similaire à celui appliqué dans la section du mean-reverting . Notre but c'est de capturer le signal inhérent au spread $Y_t - X_t$ en utilisant les termes de la signature.

Comme $d = 2$, $\ell = (\ell_1, \ell_2)$, mais pour des raisons de cohérence , on notera : $\ell = (\ell_X, \ell_Y)$ Et pour chaque $\ell_i, i \in \{X, Y\}$, ses composantes sont $\ell_i = (\ell_i^0, \ell_i^1, \ell_i^2, \ell_i^3)$.

$$\text{On a : } \hat{\mathbb{Z}}_{0,t}^{\leq 1} = \left(\underbrace{1}_{\text{associé à } \ell_i^0}, \underbrace{\begin{bmatrix} t \\ \text{associé à } \ell_i^1 \\ X_t - X_0 \\ \text{associé à } \ell_i^2 \\ Y_t - Y_0 \\ \text{associé à } \ell_i^3 \end{bmatrix}} \right)$$

$$\text{et } \forall i \in \{X, Y\}, \langle \ell_i, \mathbb{Z}_{0,t}^{\leq 1} \rangle = \ell_i^0 \cdot 1 + \ell_i^1 \cdot t + \ell_i^2 \cdot (X_t - X_0) + \ell_i^3 \cdot (Y_t - Y_0). \quad (\text{E1})$$

Intuitivement, $\xi_t^1 \approx Y_t - X_t$ et $\xi_t^2 \approx X_t - Y_t$.

Pour mieux auditer les fonctionnelles que ressort le signature trader lors du *fitting* , on s'attend à avoir :

- $\ell_i^0 \simeq 0$ car $X_0 = Y_0$
- $\ell_i^1 \simeq 0$ puisqu'il s'agit de processus stationnaires
- $\ell_i^2 + \ell_i^3 \simeq 0$ puisque nous voulons capturer le signal inhérent à $Y_t - X_t$ (développer E1 pour voir clairement).

Résultats

On suppose : $X_0 = Y_0 = 100$; $\sigma_X = 0,1$; $\sigma_Y = 0,15$; $M = 30000$ (Plus la taille de l'échantillon est grande plus le *signature trading* est efficace) , $\kappa = 150$, T = trading sur 1 mois. Ci-dessous un aperçu de la trajectoire :

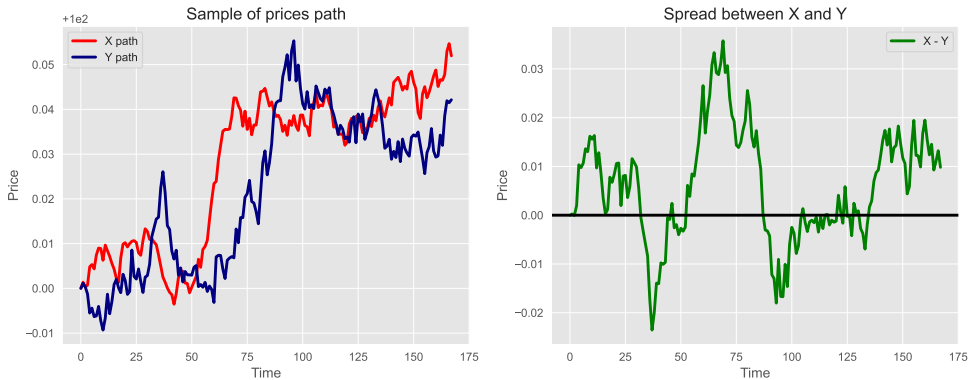


FIGURE 9 – Exemple de trajectoire de processus pairs et de leur spread.

Ci-dessous des graphiques illustratifs des résultats :

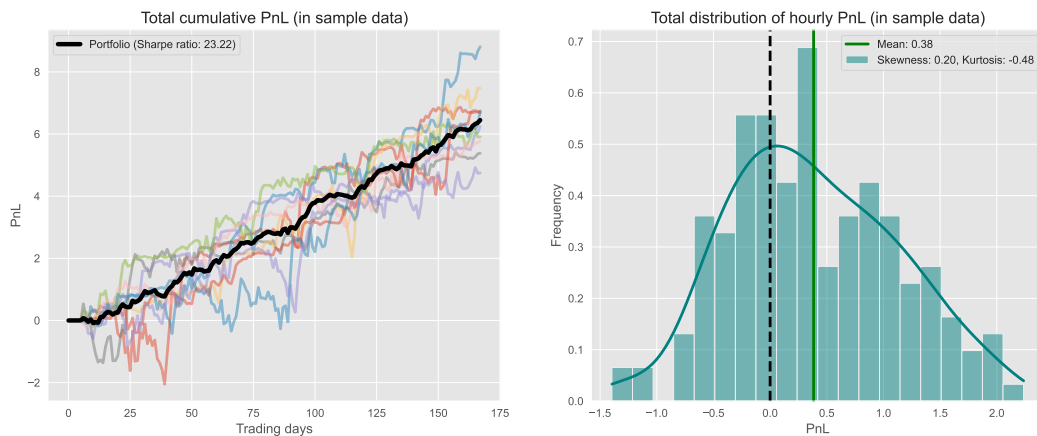


FIGURE 10 – Performance de notre *signature trading* sur les données *in-sample*.

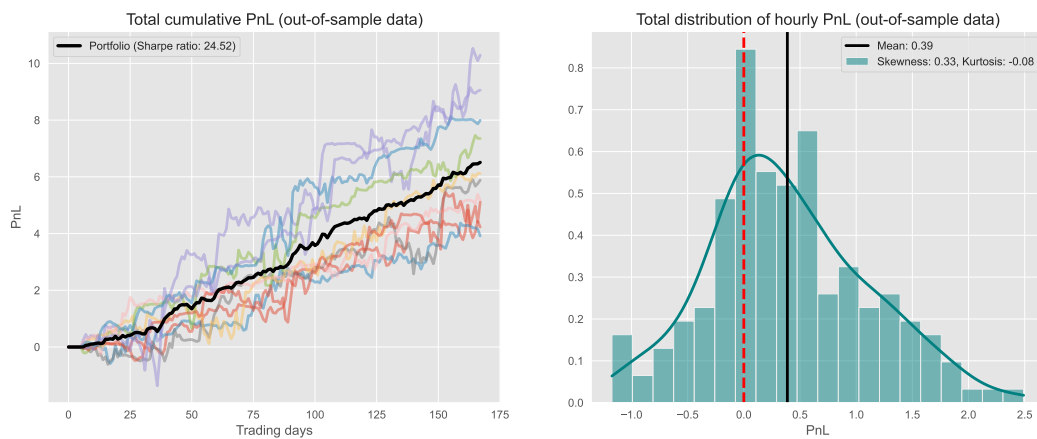


FIGURE 11 – Performance de notre *signature trading* sur les données *out-of-sample*.

$$\ell_X = \left(0.58, \begin{bmatrix} 0.001 \\ +5490 \\ -5530 \end{bmatrix} \right), \ell_Y = \left(-1.21, \begin{bmatrix} 0.004 \\ -5458 \\ +5745 \end{bmatrix} \right)$$

On constate que dans l'ensemble les fonctionnelles sont cohérentes. Toutefois, pour que les termes de fonctionnelles convergent tous vers ceux à quoi on s'attend théoriquement, il faut une grande taille d'échantillon d'entraînement.

De plus, comme pour le *mean-reverting*, notre *signature trader* s'ajuste bien tant en *in-sample* qu'en *out-of-sample*. Cela atteste de sa capacité à trader dans ces contextes.

Jusqu'ici, on s'était intéressé à des univers sans facteurs exogènes. Ce facteur sera dorénavant pris en compte.

3.4 Prise en compte de signaux exogènes

Dans ce scénario , on fait les mêmes hypothèses que dans le papier :

On suppose que le prix de notre actif X est *drivé* par une fonction $\phi(t, f_{0,t})$ de notre signal exogène $f : dX_t = \phi(t, (f_u)_{u \in [0,t]})dt + \sigma^X dW_t^X$.

Comme $\phi(t, (f_u)_{u \in [0,t]})$ est une fonction des séries temporelles passées de f , il peut être difficile pour un trader de modéliser directement la dynamique de ce système s'ils ne connaissent pas la forme explicite de ϕ .

Dans la littérature ,il est mentionné qu'en pratique , un trader peut utiliser un filtre de Kalman pour capturer l'impact d'un signal bruité . Dans notre cadre , nous proposons le système suivant pour le processus de signal f et son impact consécutif sur l'actif X

- Le signal f est modélisé par un processus d'*Ornstein-Uhlenbeck* avec moyenne nulle : $df_t = -\kappa f_t + \sigma^f dW_t^f$;
- L'information décroissante du signal entre 0 et t est modélisée par le processus Z : $Z_t = \int_0^t K(t-s)df_s$ avec $K(t,s) = \exp(-\alpha(t-s))$ le *decay kernel* dans notre cas ;
- $dX_t = Z_t dt + \sigma^X dW_t^X$.

Expérimentations numériques

On suppose : $\kappa = 4.5, \sigma_X = 0.2, \sigma_f = 0.25, \alpha = 22, X_0 = 10$, période de trading = 1 an et échantillon de taille $M = 15000$.

Ci-dessous un aperçu de quelques simulations de f, X et Z :

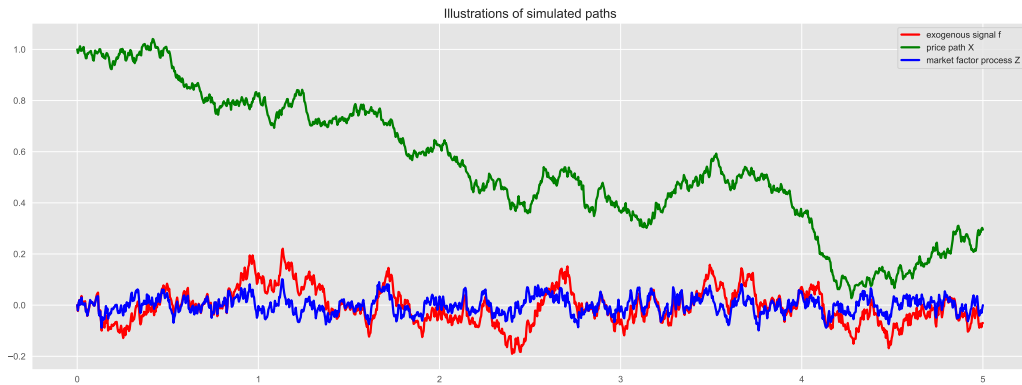


FIGURE 12 – Illustrations de trajectoires simulées.

Afin de mieux capturer la relation non linéaire entre le signal et le prix de l'actif, on priorise une signature tronquée d'ordre 2. Dans ce cadre , on a :

$$\hat{Z}_{(s,t)}^{\leq 2} = \left(1, \begin{bmatrix} t - s \\ X_t - X_s \\ f_t - f_s \end{bmatrix}, M^{\hat{Z}}(s,t) \right)$$

Avec $M^{\hat{Z}}(s,t)$, une matrice de taille 3×3 définie par :

$$M^{\hat{Z}}(s,t) = \int_{s < u_1 < u_2 < t} dZ_{u_1}^{(i)} dZ_{u_2}^{(j)}$$

Ci-dessous des graphiques illustratifs des résultats :

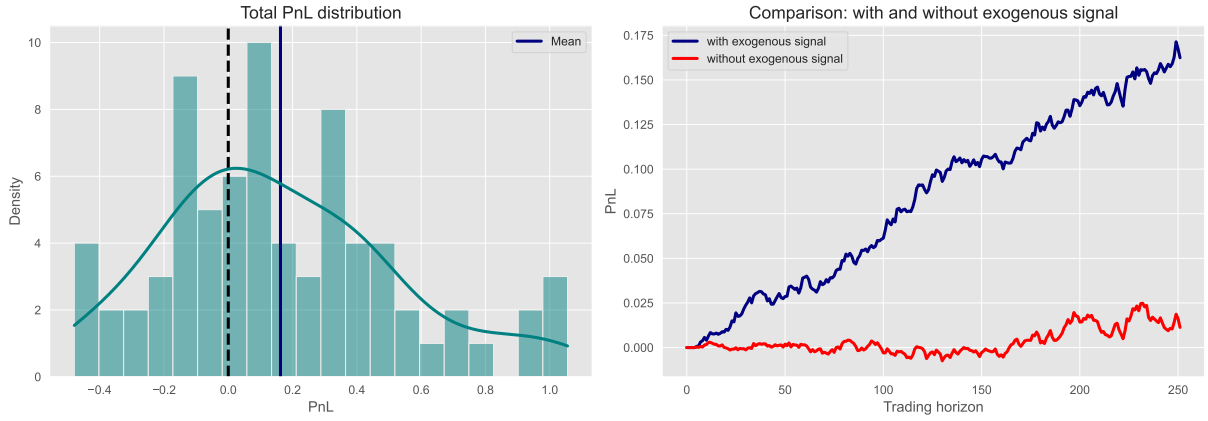


FIGURE 13 – Performance du *signature* trader avec et sans signal exogène (75 actifs).

L'ensemble des fonctionnelles que ressort notre signature trader est :

$$\left(-0.08, \begin{bmatrix} 2.e - 4 \\ 9.58 \\ \boxed{14.56} \end{bmatrix}, \begin{bmatrix} -3e - 7 & 4e - 4 & -2.e - 3 \\ -0.01 & -0.36 & 0.93 \\ \boxed{-0.02} & \boxed{0.65} & \boxed{2.88} \end{bmatrix} \right)$$

On constate sans surprise que le signal exogène a un véritable pouvoir explicatif au vu des coefficients des fonctionnelles peu importe le nombre d'actifs (on a testé 10,75,100). De plus, en testant les fonctionnelles calibré sur 75 actifs pendant une année en prenant en compte le facteur exogène, le PnL sur la période est en augmentation continue par rapport au scénario où on n'intègre pas de signal exogène où le PnL est autour de 0.

Conclusion

Contrairement à ce que stipule l'hypothèse d'efficience des marchés financiers, il existe des anomalies statistiques dans les séries temporelles que les stratégies de trading cherchent à capter, également appelées signaux ou alphas, pour en tirer profit. Cependant, les méthodes traditionnelles d'optimisation de portefeuille peinent à saisir ces signaux complexes.

Dans notre étude, nous nous sommes intéressés au papier [3] qui introduit une nouvelle classe de modèles fondée sur la signature des chemins, qui sont une alternative aux méthodes de *machine learning* grâce à leur capacité à gérer des dynamiques et des signaux non linéaires et dépendants du chemin dans un contexte d'optimisation de portefeuille. Contrairement aux méthodes d'apprentissage automatique, ce cadre ne nécessite pas optimisation par descente de gradient et fournit une solution sous forme fermée, légère à utiliser en pratique bien qu'il soit axé sur des données.

Toutefois, nos implémentations ont révélé que comme la majorité des méthodes de type *data-driven*, il faut une grande quantité de données à notre algorithme pour pouvoir s'ajuster correctement (avoir des fonctionnelles stables). Par exemple, pour un ordre de troncature égal à 1 il faut environ une taille d'échantillon $M = 1000$ et à minima $M = 10000$ pour une troncature d'ordre 2.

Il ressort que le trading par signature apparaît être une méthode de trading très prometteuse, à condition d'avoir suffisamment de puissance de calcul et de grandes quantités de données financières, si on désire avoir d'excellentes performances.

D'autres études peuvent être menées afin d'identifier mathématiquement les ordres de troncatures optimaux et comment pourrait-on utiliser cette technique des signatures dans d'autres problématiques du trading quantitatif comme l'exécution optimale [8] ou l'identification d'entrée et sortie optimales dans des positions à d'arbitrage statistique [9] ou encore de *hedging* [10].

Références

- [1] H. MARKOWITZ. « Portfolio Selection ». In : (2001).
- [2] R. CONT. « Empirical Properties of Asset Returns : Stylized Facts and Statistical Issues ». In : (1952).
- [3] M. W. OWEN FUTTER Blanka Horvath. « Signature Trading : A Path-Dependent Extension of the Mean-Variance Framework with Exogenous Signals ». In : (2023).
- [4] T. L. DANIEL LEVIN et H. NI. « Learning from the past, predicting the statistics for the future, learning an evolving system. » In : *url : <http://arxiv.org/abs/1309.0260>* (2013).
- [5] T. CASS et W. F. TURNER. « Topologies on unparameterised path space , *url : <http://arxiv.org/abs/2206.11153>* ». In : (2022).
- [6] B. HAMBLBY et T. LYONS. « Uniqueness for the signature of a path of bounded variation and the reduced path group ». In : (2006).
- [7] B. H. GUY FLINT et T. LYONS. « “Discretely sampled signals and the rough Hoff process” ». In : *Stochastic Processes and their Applications 126.9* , *pp. 2593– 2614*. (2016).
- [8] I. P. A. JASDEEP KALSI Terry Lyons. « Optimal execution with rough path signatures ». In : (2019).
- [9] K. L. BOMING NING Prakash Chakraborty. « Optimal Entry and Exit with Signature in Statistical Arbitrage ». In : (2024).
- [10] S. N. TERRY LYONS et I. P. ARRIBAS. « Nonparametric pricing and hedging of exotic derivatives ». In : (2019).