Analysis and modeling of high-frequency financial market data using Hawkes processes and neural networks

Ruihua Ruan

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Université Paris Dauphine

Analyse et modélisation des données à haute fréquence sur les marchés financiers en utilisant les processus de Hawkes et les réseaux de neurones

Soutenue par

Ruihua Ruan

Le 4 Décembre 2023

École doctorale n°543

Ecole Doctorale SDOSE

Spécialité

Mathématiques

Composition du jury :

Marc Hoffmann	Professeur, Université Paris Dauphine-PSL	*Président*
Samuel Cohen	Professeur, University of Oxford	*Rapporteur*
Fabrizio Lillo	Professeur, Università di Bologna	*Rapporteur*
Stéphane Gaïffas	Professeur, Université Paris Diderot	*Examinateur*
Sophie Laruelle	Maître de Conférences, Université Paris-Est Créteil	*Examinatrice (Absente)*
Ioane Muni Toke	Maître de Conférences, CentraleSupélec, Université Paris-Saclay	*Examinateur*
Emmanuel Bacry	Directeur de recherche CNRS, Université Paris Dauphine-PSL	*Directeur de thèse*
Jean-François Muzy	Directeur de recherche CNRS, Université de Corse	*Co-directeur de thèse*
Thomas Deschatre	Chercheur-Expert, EDF	*Invité*

Dauphine | PSL UNIVERSITÉ PARIS

Dauphine | PSL UNIVERSITÉ PARIS CEREMADE UMR CNRS 7534

Doctoral School **SDOSE Sciences de la Décision, des Organisations, de la Société et de l'Echange**

Research Unit **UMR 7534 - Centre de Recherche en Mathémati ques de la Décision**

Thesis presented by **Ruihua Ruan**

Defended on December 4, 2023

A thesis presented for the degree of Doctor of Philosophy

Specialty : **Mathematics**

Analysis and modeling of high-frequency data in financial markets using Hawkes processes and neural networks

Committee members

President	Marc Hoffmann	Professor, Université Paris Dauphine-PSL
Reviewers	Samuel Cohen	Professor, University of Oxford
	Fabrizio Lillo	Professor, Università di Bologna
Examiners	Stéphane Gaïffas	Professor, Université Paris Diderot
	Sophie Laruelle	Associate Professor, Université Paris-Est Créteil (Absent)
	Ioane Muni Toke	Associate Professor, CentraleSupélec, Université Paris-Saclay
Supervisors	Emmanuel Bacry	CNRS Senior Research Fellow, Université Paris Dauphine-PSL
	Jean-François Muzy	CNRS Senior Research Fellow, Université de Corse

To my parents

REMERCIEMENTS

Je voudrais tout d'abord remercier mes directeurs de thèse, Emmanuel Bacry et Jean-François Muzy. Merci pour vos conseils éclairés, vos connaissances approfondies, ainsi que pour vos intuitions et votre expérience. Je vous suis particulièrement reconnaissante pour la confiance que vous avez placée en moi. Grâce à votre encadrement, j'ai pu explorer un éventail varié de domaines scientifiques et développer des compétences théoriques et pratiques essentielles.

I would like to thank Samuel Cohen and Fabrizio Lillo for agreeing to review my thesis. Your careful reading, insightful comments, and constructive remarks have been very valuable to me. Merci à Marc Hoffmann, Sophie Laruelle, Stéphane Gaïffas, Ioane Muni Toke d'avoir accepté de faire partie de mon jury de thèse. Je tiens particulièrement à remercier Marc Hoffmann. Merci pour ton soutien constant et tes encouragements, que ce soit après les réunions ou lors de pauses café. Mes remerciements vont également à Thomas Deschatre pour le temps et l'énergie qu'il a consacrés à notre projet et pour l'aide qu'il m'a apportée chaque fois que j'en avais besoin. Travailler avec vous deux a été une expérience particulièrement enrichissante.

Je remercie à nouveau Emmanuel Bacry pour son appui financier dans le cadre de sa chaire PRAIRIE (ANR-19-P3IA-0001). Grâce à ce financement, j'ai pu réaliser ce doctorat dans les meilleures conditions et participer à de nombreux événements scientifiques. Un grand merci également à Euronext Paris pour les données qu'ils nous ont mises à disposition.

J'adresse mes remerciements à Thomas Duleu et Gilles Barès pour leur soutien considérable dans l'utilisation des outils informatiques et aussi à Anne-Laure Chagnon, César Faivre et Isabelle Bellier pour leur précieux travail administratif. En particulier, Isabelle, merci pour ta gestion des missions et pour ta gentillesse. Merci également à Beatrice Baéza pour sa disponibilité et son efficacité.

Je souhaiterais également exprimer ma sincère reconnaissance à divers chercheurs du CEREMADE qui m'ont apporté beaucoup d'aide tout au long de ces quatre années. Un merci à Alessandra, ma mentore, pour m'avoir encouragée de manière régulière. Merci à Julien, Zhenjie et François, pour nos discussions instructives. En particulier, je voudrais remercier Yating, qui est également une mentore pour moi. À chaque fois que j'avais besoin d'aide, tu étais toujours présente (je suis consciente que tu as sacrifié bon nombre de pauses café pour m'encourager et me motiver). Enfin un grand merci à Béatrice et Madalina pour votre aide précieuse et vos encouragements constants.

Un grand merci à ceux qui ont partagé le bureau avec moi. Merci à Jean, Lucas et Luke pour ces bons moments partagés et pour votre soutien. J'ai beaucoup apprécié les discussions et les moments

sportifs. Je tiens particulièrement à remercier Luke pour son aide dans la relecture de cette thèse. Mes remerciements s'adressent aussi à Alexandre, Carmela, Elena, David, Fabio, Florin, Grégoire, Laura, Louis, Peter, Quentin, Richard, Thinh. Grâce à chacun de vous, mon expérience aux bureaux C606 et B220 restera un excellent souvenir.

Merci à Peng, j'ai pu trouver cette thèse grâce à toi. À Charly, merci pour ton assistance sur VS Code qui a vraiment boosté ma productivité. Merci aussi à tous les autres doctorant.e.s : Adrien (les deux), Adéchola, Antoine (les deux), Changqing, Claudia, Danièle, Donato, Grégoire, Kathi, Kexin, Lorenzo, Luca, Othmane, Quan, Umberto, Théo, Xiaozhen et tous les autres que je n'ai pas cités.

Ces quatre ans auraient été beaucoup moins amusants sans le sport. Un merci à mes amis de sport à Dauphine, que ce soit pour le yoga (Jean, Kathi, Luke, Quan), l'escalade (Jean, Kathi, Louise, Luke, Théo), la natation (Damien, Souad), ou encore la course à pied (notre équipe papaya) ! En particulier, je tiens à exprimer mes remerciements à ma professeure de yoga, Souad, qui m'a appris à m'accepter telle que je suis.

En dehors du CEREMADE, j'ai également eu la chance de passer du temps avec le laboratoire FiME grâce à Damien et Thomas. J'ai pu profiter des événements organisés par FiME, des séminaires, des écoles d'été, etc. Merci à toute l'équipe, en particulier Clémence, Damien, Olivier et Thomas pour ces moments précieux.

Un grand merci à mes amis chinois à Paris, Cheng, Ruodan, Boyuan et Shiwen, qui ont ajouté une touche spéciale à ces quatre années. À Cheng et Ruodan, merci pour votre amitié indéfectible. Depuis notre arrivée en France il y a huit ans, nous ne nous sommes jamais quittées. Mes remerciements vont également à mes amis du *French Terrors* du Michigan, pour les activités récréatives le week-end.

Loin de ma famille en Chine pendant ces huit ans, c'est grâce à mes familles françaises que je me sens moins étrangère. Merci à mon parain Robert et à Catherine. Votre aide et vos nombreux conseils ont grandement facilité mon intégration en France. Merci à la famille d'Olivier. Grâce à vous, les fêtes françaises sont devenues de véritables célébrations familiales pour moi, plutôt que de simples jours fériés.

Now, I would like to thank my dear family in China (now with a part in the US), thank you for giving me unlimited love and support. To my parents, it was your dedication to my education that has made it possible for me to achieve my dream. Your company and protection are the constant sources of my happiness. To my brothers and my sister, you're the ones who introduced me to the outside world, who gave me ambition and helped me widen my horizon. Especially, I would like to thank my sister, for always being an example for me and offering me strong support. I also extend my thanks to my brother, sister-in-law, and adorable nephew Jiyong in the US, who brought me much joy the last winter holidays and continue to brighten my weekends.

Enfin, un immense merci à Olivier. Être avec toi est ma plus grande récompense en France. Merci pour ta présence, ton écoute et ta patience infinie. Ces quatres années n'auraient jamais été les mêmes sans toi à mes côtés.

RESUMÉ

Titre : Analyse et modélisation des données à haute fréquence sur les marchés financiers en utilisant les processus de Hawkes et les réseaux de neurones

Résumé : Cette thèse est consacrée à l'étude de la microstructure du marché dans les marchés électroniques, en mettant l'accent sur deux sujets clés. Le premier sujet concerne la construction de deux modèles pour les événements de Niveau 1 dans le carnet d'ordres, en utilisant des approches basées sur des modèles statistiques. Le premier modèle consiste en un processus de Hawkes non-linéaire pour modéliser la dynamique du *bid-ask spread*, appelé le modèle "*State-Dependent Spread Hawkes*". En intégrant les tailles des sauts du *spread* et sa valeur dans la fonction d'intensité, ce modèle est capable de capturer diverses propriétés statistiques du *spread*. Le second modèle, appelé "*Hawkes process with shot noise*", est utilisé pour séparer les sources de corrélation endogènes et exogènes entre deux prix d'actifs. Pour ce faire, ce modèle suppose l'existence d'un processus latent (*shot noise*), représentant des comportements d'agents spécifiques non directement observables sur le marché. Théoriquement, des théorèmes de limite sont démontrés et dans la pratique, l'estimation est facilitée par une technique d'estimation non paramétrique.
Le second sujet concerne l'analyse et la caractérisation des comportements des agents sur le marché financier, en utilisant des approches basées sur des réseaux neuronaux profonds. Ce sujet comprend deux tâches. La première tâche consiste à classifier les agents en fonction de leurs ordres passés, grâce à une approche d'apprentissage supervisé. La deuxième tâche vise à apprendre la représentation des comportements des agents, en utilisant un modèle d'apprentissage auto-supervisé fondé sur la *triplet loss*. Ces représentations apprises nous permettent d'appliquer l'algorithme de clustering K-means pour identifier des types de comportements distincts au sein de chaque groupe et ainsi analyser les comportements des agents.

Mots clés : Microstructure du marché, Carnet d'ordres, Processus de Hawkes, Réseaux neuronaux

ABSTRACT

Title: Analysis and modeling of high-frequency data in financial markets using Hawkes processes and neural networks

Abstract: This thesis is devoted to the study of market microstructure in electronic markets, focusing on two key topics. The first topic concerns the construction of two models for Level 1 events in Limit Order Book, using model-driven approaches. The first model is a non-linear Hawkes process for modeling spread dynamics, referred to as the "State-Dependent Spread Hawkes" model. This model, integrating spread jump sizes and spread state into intensity, can capture a range of statistical properties of the spread. The second model, called the "Hawkes process with shot noise" model, is used to disentangle the endogenous and exogenous sources of correlation between two asset prices. To do so, this model assumes the existence of a latent shot noise process, representing specific agent behaviors not directly observable in the market. Theoretically, limit theorems are demonstrated and in practice, the estimation is facilitated through a non-parametric technique.
The second topic involves analysis and characterization of agent behaviors in the financial market, by employing data-driven approaches that relies on deep neural networks. This topic includes two tasks. The first task is to classify agents, based on their placed orders, through a supervised learning approach. The second task is to learn the representation of agents' behaviors, using a self-supervised learning model based on Triplet loss. These learning representations allow us to apply the K-means clustering algorithm to identify distinct behavior types within each cluster and therefore analyze the behaviors of agents.

Keywords: Market microstructure, Limit Order Book, Hawkes processes, Neural networks

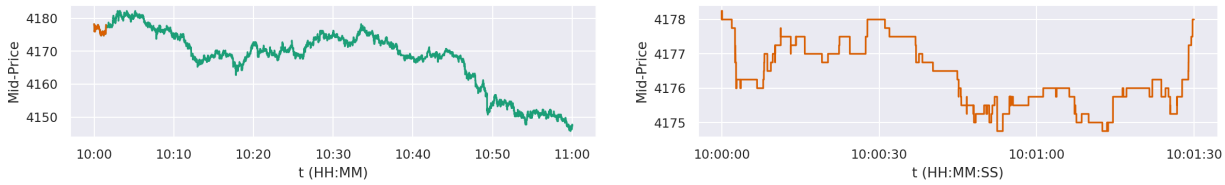
CONTENTS

INTRODUCTION GÉNÉRALE

I - Contexte et motivations

Un marché financier est un système complexe qui est composé de nombreux éléments en interaction : il comprend des investisseurs individuels et institutionnels, des entreprises, des gouvernements et des régulateurs. Pendant longtemps, la modélisation de la dynamique des marchés financiers, en particulier l'évolution des prix, a été un défi fondamental pour les mathématiques financières. La capacité à modéliser avec précision et prédire le comportement du marché est cruciale pour élaborer des stratégies d'investissement et gérer les risques.

À l'échelle macroscopique, les processus de prix apparaissent comme des trajectoires échantillonnées à partir de modèles classiques en mathématiques financières, tels que le mouvement brownien (voir Fig. 1a). Cependant, lorsque l'on se penche sur l'échelle microscopique et que l'on examine les mouvements de prix sur de très courtes périodes de temps (par exemple, des secondes), le comportement des processus financiers devient assez différent. Ce comportement distinct est illustré dans la Figure 1.1b, qui se focalise sur un processus échantillonné à 90 secondes. En raison de la structure des ticks, le prix prend des valeurs discrètes et se manifeste sous forme de processus de sauts. Ce phénomène est appelé l'effet de microstructure. Alors que l'échelle macroscopique de la modélisation nous donne une vision globale de la tendance du marché, l'échelle microscopique fournit un aperçu de la microstructure du marché, ce qui est essentiel pour comprendre et décrire son comportement macroscopique.



(a) À l'échelle macroscopique. Le segment orange représente le processus initial de 90 secondes.

(b) À l'échelle microscopique. La trajectoire se concentre sur le processus initial de 90 secondes dans la Figure 1.1a.

FIGURE 1 – Exemples de processus de *mid-price* du CAC40 Future.

Le concept de théorie de la microstructure du marché a émergé à la fin du 20e siècle et a connu une évolution significative depuis 2005. Cette théorie étudie le processus de formation des prix dans le cadre des mécanismes de négociation spécifiques, de la divulgation des informations de négociation et des composants du marché. Elle évalue l'efficacité et l'équité du marché à l'aide d'indicateurs typiques tels que la liquidité et la découverte des prix. On peut trouver un panorama complet des aspects théoriques et empiriques de la microstructure des marchés dans des ouvrages tels que Bouchaud et al. (2018); Harris (2003); Hasbrouck (2007); Lehalle and Laruelle (2018); O'hara (1998). En raison du développement rapide du trading algorithmique et électronique, la microstructure des marchés demeure l'un des domaines qui evolue le plus rapidement en recherche financière. Un nombre croissant d'études est désormais consacré à l'analyse et à la modélisation des flux d'ordres et de la dynamique des prix au niveau microscopique.

Pour décrire la microstructure des flux d'ordres et la dynamique des prix, les processus ponctuels sont souvent utilisés (Abergel and Jedidi, 2013; Daniels et al., 2003; Smith et al., 2003). Dans ce contexte, comme dans Bacry et al. (2015), le processus de Hawkes est une classe très populaire de modèles qui a fait ses preuves dans la déscription des propriétés dynamiques de différentes quantités des carnets d'ordres (Bacry et al., 2013a; Bowser, 2007; Large, 2007; Toke, 2011). Introduits par Hawkes (1971a,b), les processus de Hawkes sont des processus ponctuels auto-excitants capables de capter l'effet d'excitation mutuelle entre les événements. Ils ont gagné en popularité en raison de leur capacité à rendre compte des interactions entre les événements et de leur aptitude à fournir une interprétation simple et convaincante de ces interactions.

Dans cette thèse, notre objectif est d'examiner et de développer des modèles pour comprendre la microstructure complexe des données à haute fréquence au sein d'un vrai marché financier. Nos recherches sont facilitées par une grande base de données généreusement mise à disposition par Euronext Paris. Cette base de données comprend un historique complet de toutes les commandes passées dans le carnet d'ordres pour 40 actions de l'indice CAC40, ainsi que le contrats Future CAC40. En particulier, chacune de ces commandes est identifiée par le numéro d'identification anonyme de l'agent qui l'a déposée. Cette caractéristique nous permet d'étudier le comportement des agents individuels sur le marché.

Plus précisément, nous nous concentrerons sur les questions suivantes.

Comment le marché peut-il résister aux chocs de liquidité ? (Chapitre 3)

La liquidité d'un actif désigne la facilité avec laquelle il peut être converti en espèces sans affecter significativement le prix. Elle revêt une grande importance dans l'investissement car elle reflète la solidité du marché. Cependant, il n'est pas toujours facile de donner une définition précise de la liquidité. Dans une perspective à court terme, le *bid-ask spread*, défini comme l'étendue de l'écart entre les cours d'achat et de vente, peut être considéré comme une approximation de la liquidité du marché. Lorsque ce *spread* est important, cela indique un manque de liquidité et un déséquilibre sur le marché. En revanche, un spread faible signale un équilibre. Si le spread se creuse rapidement et de manière importante, cela peut déclencher une crise de liquidité (Díaz and Escribano, 2020; Fong et al., 2017; Fosset et al., 2020a; Goyenko et al., 2009).

Dans le **Chapitre 3**, nous cherchons à comprendre comment un marché peut résister aux chocs ; autrement dit, comment un marché peut "se réparer" lorsque le spread varie. Nous proposons un modèle de processus de Hawkes pour décrire la dynamique du spread. Pour commencer, nous décomposons les processus de spread en deux processus croissants, chacun associé à des événements qui amplifient ou diminuent le spread. Cette décomposition transforme le processus de spread en

un processus de comptage bidimensionnel.

Les processus de Hawkes ont prouvé leur efficacité pour modéliser l'effet d'excitation mutuelle entre les flux de commandes, les mouvements de prix, etc. (Abergel and Jedidi, 2015; Bacry et al., 2013a, 2015). Cependant, modéliser le spread par des processus de Hawkes classiques n'est pas direct en raison de la contrainte selon laquelle le spread doit être strictement positif. Pour répondre à cette problématique, Zheng et al. (2014) et Fosset et al. (2020a) se sont intéressés à des processus de Hawkes non linéaires qui imposent une intensité nulle pour les événements de diminution du spread, lorsque celui-ci atteint son minimum, soit un.

Dans cette thèse, inspirés par les modèles *Queue-reactive* (Huang et al., 2015; Wu et al., 2019), nous proposons un nouveau modèle appelé le *State-Dependent Spread Hawkes* (SDSH). Ce modèle peut être considéré comme une généralisation des modèles discutés dans Zheng et al. (2014) et Fosset et al. (2020a). Notre modèle SDSH intègre non seulement la mémoire des événements passés, mais également l'état actuel du spread dans la fonction d'intensité. Dans le Chapitre 3, nous démontrerons en quoi ce modèle SDSH permet de capturer davantage de propriétés statistiques.

Quels facteurs microstructurels contribuent aux corrélations entre les prix de différentes actions ? (Chapitre 4)

La question précédente se concentrait sur la modélisation d'un actif unique. Dans cette question, nous considérons plusieurs actifs et les corrélations de leurs prix. La corrélation est une mesure statistique qui détermine comment les actifs évoluent les uns par rapport aux autres. Elle est couramment utilisée en finance pour comprendre le comportement global du marché ou pour évaluer le potentiel de diversification d'un portefeuille.

Les actifs peuvent présenter une corrélation en raison de divers facteurs. L'une des sources les plus courantes de corrélation est leur secteur ou leur industrie. Par exemple, BNP Paribas et Société Générale sont toutes deux des banques, elles sont donc susceptibles d'être affectées par des conditions macroéconomiques similaires, telles que les taux d'intérêt, l'inflation et le comportement des consommateurs. Par conséquent, leurs cours d'actions peuvent évoluer dans la même direction.

En plus de l'affiliation à un secteur ou à une industrie, les événements mondiaux peuvent également avoir un impact sur plusieurs entreprises et entraîner des corrélations entre les secteurs. Ces événements comprennent les communiqués de presse, les interventions gouvernementales, les catastrophes naturelles comme la Covid-19 et d'autres chocs géopolitiques ou macroéconomiques.

Pour certaines valeurs mobilières spécifiques comme les contrats à terme ou les options, l'actif sous-jacent commun de deux valeurs mobilières peut être la source la plus importante de leur corrélation. Par exemple, Bobl et Bund sont fortement corrélés car ils représentent des contrats sur le même actif sous-jacent avec des échéances différentes (voir Figure 6 dans Bacry et al. (2013a)).

Dans le **Chapitre 4**, nous souhaitons aborder la question de l'origine microscopique des corrélations. À cette fin, nous introduisons des modèles qui englobent différentes sources de corrélation. La première source est la source endogène, résultant des mécanismes de rétroaction internes des processus de prix eux-mêmes. La deuxième source est une source exogène, induite par des facteurs externes, tels que les nouvelles et le comportement des agents. La première source a été étudiée dans Bacry et al. (2013a). Les auteurs ont également établi certains théorèmes limites pour le modèle de processus de Hawkes (Bacry et al., 2013b).

Dans cette thèse, nous proposons une version étendue du modèle de processus de Hawkes introduit

par Bacry et al. (2013a), que nous appelons le modèle *Hawkes with shot noise*. En étendant le modèle classique des processus de Hawkes pour les prix, nous introduisons une dimension latente supplémentaire pour capturer la source exogène de corrélation. (Dans ce contexte, "latente" indique que les événements dans cette dimension ne sont pas observables). Cette dimension est un processus de Poisson qui influence les prix des deux actifs. La logique derrière l'ajout de cette dimension latente est directe : si les prix de deux actifs commencent soudainement et simultanément à évoluer de manière significative, une explication plausible est que ces fluctuations sont en effet entraînées par des éléments externes communs.

L'inclusion de la dimension latente rend le modèle plus réaliste, mais introduit davantage de défis en termes d'estimation. La méthode des cumulants non paramétriques (NPHC), introduite par Achab et al. (2017), s'avère tout aussi efficace pour le modèle de *Hawkes with shot noise*. Le Chapitre 4 présente une validation empirique de l'efficacité de cette méthode.

Quels rôles les participants jouent-ils sur le marché ? (Chapitres 5 et 6)

Pour les deux questions précédentes, nous considérions le marché comme une entité auto-régulatrice. Cependant, le marché est un système complexe composé de nombreux agents. Les performances globales du marché découlent des actions individuelles de chaque agent sur celui-ci. Par conséquent, comprendre les rôles et les contributions de ces agents est crucial pour comprendre le fonctionnement du marché dans son ensemble.

À ce stade, nous introduisons la méthodologie de modélisation basée sur les agents (ABM) qui est extrêmement utile dans différents domaines. Elle peut simuler les actions et les interactions des individus et des organisations de manière complexe et réaliste (Axtell and Farmer, 2022; Iori and Porter, 2018). Cependant, en raison de préoccupations concernant la protection des données, seules quelques études (Cartea et al., 2023; Cont et al., 2023; Rambaldi et al., 2019) ont pu accéder aux données avec l'identification des membres. Dans cette thèse, grâce à l'accès à la base de données d'Euronext Paris, qui contient l'identification des agents, nous pourrons étudier le comportement de trading de chaque agent individuel.

En examinant les actions d'un agent sur une période donnée, nous pouvons obtenir des informations sur certains aspects macroscopiques : l'agent a-t-il tendance à agir en tant que *market maker* (teneur de marché) ou *market taker* (preneur de marché) ? Ont-ils une stratégie de trading distincte ? Comment se manifeste l'évolution de leurs stratégies au fil du temps ? Répondre à ces questions est très important à la fois pour l'exploration académique et la régulation du marché. Ces réponses peuvent être apportées par des modèles basés sur les agents, pour améliorer les caractéristiques microstructurelles du marché. De plus, elles peuvent également aider les régulateurs à identifier les comportements irréguliers sur le marché.

Les chapitres 5 et 6 seront consacrés à répondre aux questions mentionnées ci-dessus. Nous caractérisons un agent à un moment donné par une séquence d'ordres consécutifs qu'il exécute. Le chapitre 5 peut être considéré comme une étape fondamentale. En utilisant la méthode d'apprentissage supervisé pour identifier les agents, nous étudions l'importance des variables et les performances de la classification. Les méthodes d'apprentissage profond ont prouvé leur efficacité dans la modélisation des carnets d'ordres limites dans des travaux antérieurs Sirignano and Cont (2019); Sirignano (2019); Zhang et al. (2019). Ce chapitre démontrera leur efficacité pour caractériser les agents.

Dans le chapitre 6, nous approfondissons la réponse aux questions précédentes. Ici, nous introduisons d'abord les séquences d'ordres des agents dans une tâche prétexte, dans le but d'apprendre une

représentation du comportement d'un agent à un moment donné. La tâche prétexte utilise une approche d'apprentissage contrastif auto-supervisé avec une perte de triplet (Schroff et al., 2015). De façon similaire aux travaux sur le traitement du langage naturel, tels que le succès remarquable de Mikolov et al. (2013b), les représentations apprises peuvent révéler une structure intrinsèque au sein des séquences d'ordres. Par conséquent, des algorithmes de regroupement peuvent être ensuite appliqués pour regrouper les agents ayant des comportements similaires. Nous démontrons que les agents peuvent être regroupés en groupes caractérisés par des stratégies de trading distinctes.

Les sections suivantes sont consacrées à un résumé des principaux résultats de cette thèse.

II - Résumé des principaux résultats

II.1 Résumé du Chapitre 3

Le **chapitre 3** correspond au papier Ruan et al. (2023b), soumis à la revue *Market Microstructure and Liquidity (MML) Journal*. Dans ce chapitre, nous proposons un modèle de processus de Hawkes non linéaire pour modéliser la dynamique du spread entre les cours acheteur et vendeur, que nous appelons le modèle *State-Dependent Spread Hawkes* (SDSH).

Le *bid-ask spread* est défini comme la différence entre le prix de vente le plus bas et le prix d'achat le plus élevé dans un carnet d'ordres. Il est généralement utilisé comme mesure de la liquidité du marché et joue un rôle crucial dans les analyses financières. Dans ce travail, nous représentons un processus de *spread* sous la forme de $(S_t)_{t\geqslant 0}$, qui peut être décomposé en deux termes S_t^+ et S_t^-, représentant respectivement les sauts positifs et négatifs du spread. Pour l'instant, tous les sauts sont supposés être de taille d'un tick, ce qui donne $S_t = S_0 + S_t^+ - S_t^-$. Avant d'introduire notre modèle SDSH, examinons de plus près deux modèles existants étroitement liés à notre approche. Le premier modèle est le modèle de spread proposé par Zheng et al. (2014), qui est un modèle de Hawkes contraint avec les fonctions d'intensité suivantes :

$$\begin{aligned} \lambda_t^+ &= \mu^+ + \sum_{e\in\{+,-\}} \int_0^t \varphi^{+,e}(t-s)\, dS_s^e, \\ \lambda_t^- &= 1_{S_{t-}\geqslant 2}(\mu^- + \sum_{e\in\{+,-\}} \int_0^t \varphi^{-,e}(t-s)\, dS_s^e), \end{aligned} \tag{II.1}$$

où λ^+ (resp. λ^-) est l'intensité associée à S^+ (resp. S^-). La condition $1_{S_{t-}\geqslant 2}$ garantit que le spread reste strictement positif en tout temps.

Le deuxième modèle, exploré dans Fosset et al. (2020a), est défini comme suit :

$$\begin{aligned} \lambda_t^+ &= \mu^+ + \int_0^t \alpha\beta e^{-\beta(t-s)}\, dS_s^+ \\ \lambda_t^- &= \mu^- 1_{S_{t-}\geqslant 2} \end{aligned} \tag{II.2}$$

Ce modèle est un exemple spécifique de (II.1) lorsque $\varphi^{-,+}$ et $\varphi^{-,-}$ sont fixés à zéro. Les auteurs démontrent que lorsque $\alpha < 1 - \frac{\mu^+}{\mu^-}$, le système de Hawkes dans (3.1.2) est stable et le processus de spread est stationnaire.

Le modèle SDSH peut être considéré comme une extension des deux modèles précédents. Nous l'étendons (II.1) sous deux aspects différents :

- Les tailles des sauts ne sont plus contraintes à être d'une unité de tick. Au lieu de cela, nous considérons la possibilité de K tailles de sauts différentes. Cela fait que notre modèle prend la forme d'un processus de Hawkes à $2K$ variables. La valeur de K est un hyperparamètre qui peut être choisi de manière flexible en fonction de données spécifique.
- La contrainte $1_{S_{t-}\geqslant 2}$ pour λ^- est remplacée par des fonctions positives ou nulles plus générales $f(S_{t-})$. Ces nouvelles fonctions f, applicables à la fois à λ^+ et à λ^-, permettent au modèle d'incorporer le fait bien connu que le spread a tendance à revenir à sa moyenne. Par conséquent, un terme "dépendant de l'état" est introduit. Ces fonctions f peuvent être calibrées à l'aide de données disponibles.

En résumé, nous notons S_t^e le processus de comptage qui comptabilise le nombre de sauts de taille e, pour $e \in \mathcal{E} := \{+1, +2, \dots, +K, -1, -2, \dots, -K\}$. Le processus de spread S_t peut être exprimé comme suit :

$$S_t = S_0 + \sum_{k=1,2,\dots,K} k S_t^{+k} - \sum_{k=1,2,\dots,K} k S_t^{-k}.$$

Soit λ^e la fonction d'intensité du processus de comptage S^e. Le modèle SDSH est formulé comme suit :

$$\lambda_t^e = f^e(S_{t-}) \left[\mu^e + \sum_{e' \in \mathcal{E}} \int_0^t \varphi^{e,e'}(t-s)\, dS_s^{e'} \right]$$

Ici, $f^{-k}(s)$ doit être égal à 0 lorsque $s \leqslant k$, afin de maintenir le spread positif. Nous supposons que les noyaux sont paramétrés comme une somme de L termes exponentiels, donnés par :

$$\varphi^{e,e'}(t) = \sum_{l=1}^{L} \alpha_l^{e,e'} \beta_l e^{-\beta_l t}$$

Propriété de Markov et ergodicité

Soit $X_t^{e,e'} := \int_0^t \varphi^{e,e'}(t-s)\, dS_s^{e'}$, le processus combiné (S_t, X_t) est un processus markovien.

Dans un scénario simplifié où $K = 1$ et $L = 1$, ce qui signifie $\mathcal{E} = \{-1, 1\}$ et $\varphi_{e,e'}(t) = \alpha^{e,e'} \beta e^{-\beta t}$, nous pouvons énoncer la proposition suivante :

Proposition. *Le processus (S_t, X_t) est **V-uniformément ergodique** sous les conditions suivantes :*

$$\boxed{\begin{array}{l} f^-(1) = 0 \\ f^-(S) \geqslant \gamma S \text{ pour un certain } \gamma > 0 \text{ lorsque } S \geqslant 2 \\ \sup_S f^+(S)(\alpha^{+,-} + \alpha^{+,+}) < 1 \end{array}} \tag{II.3}$$

La preuve de cette proposition peut être trouvée dans 3.A.

Simulation et estimation

En utilisant la méthode classique de *thinning* introduite par Lewis and Shedler (1979); Ogata (1981) et la bibliothèque open source Tick (Bacry et al., 2017), la simulation est directe. À des fins

d'estimation, la méthode d'estimation du maximum de vraisemblance (EMV) est utilisée. Considérons une réalisation sur $[0, T]$ et notons t_k^e les instants des événements sur S^e. La fonction de log-vraisemblance peut être exprimée comme suit (pour simplifier, nous supposons que $L = 1$) :

$$\begin{aligned}\mathcal{L}(\alpha, \mu, f) &= \sum_{e\in\mathcal{E}}(-\int_0^T \lambda^e(t)\,dt + \int_0^T \log\lambda^e(t)\,dS_t^e) \\ &= \sum_{e\in\mathcal{E}}\sum_{k=1}^{S^e(T)} \log(\mu^e + \sum_{e'\in\mathcal{E}} \alpha^{ee'}\beta\int_0^{t_k^e} e^{-\beta(t_k^e - s)}\,dS_s^j) + \sum_{e\in\mathcal{E}}\sum_{k=1}^{S^e(T)} \log f^e(S_{t_k^e}) \\ &\quad - \sum_{e\in\mathcal{E}}\int_0^T (\mu^e + \sum_{e'\in\mathcal{E}} \alpha^{ee'}\beta\int_0^t e^{-\beta(t-s)}dS_s^{e'})f^e(S_t)\,dt\end{aligned}$$

Voici les hyperparamètres et leurs réglages correspondants :

- K : la plus grande taille de saut autorisée par le modèle.
- L et $\{\beta_l\}_{l=1,\dots,L}$: en pratique, choisir des β_l espacés de manière logarithmique suffit à capturer un large éventail de comportements, comme par exemple $\beta_l = \beta_1 10^{l-1}$.
- $f^e(s)$: nous supposons que toutes les fonctions $f^e(s)$ sont constantes pour s dépassant une valeur fixe $\bar{S}$. Ainsi, pour chaque e et $s \leqslant \bar{S}$, $f^e(s)$ est traité comme un paramètre.

Une illustration des résultats d'estimation sur des données simulées est disponible dans la Figure 3.1 dans le corps principal de cette thèse.

Résultats empiriques

Nous calibrons le modèle SDSH en utilisant les données CAC40 d'Euronext. Les données correspondent aux processus de spread pour 3 actions, à savoir AXA, BNP Paribas, Nokia, ainsi que le Future CAC40, sur environ 100 jours. Les hyperparamètres sont indiqués dans le Tableau 3.2.

Les Figures 2 et 3 fournissent des exemples de résultats d'estimation pour f^e (Nokia) et les noyaux essentiels $\varphi^{e,e'}$ (AXA). Comme prévu, nous observons que $f^{+1}(s)$ et $f^{+2}(s)$ présentent une tendance à la baisse et se rapprochent de 0 à mesure que s augmente. À l'inverse, $f^{-1}(s)$ et $f^{-2}(s)$ sont des fonctions globalement croissantes. Lorsque le spread est élevé, de petites valeurs de $f^{+1}(s)$ et $f^{+2}(s)$ inhibent les sauts positifs tandis que de grandes valeurs de $f^{-1}(s)$ et $f^{-2}(s)$ encouragent les sauts négatifs. Ce mécanisme renforce la tendance de retour à la moyenne du spread.

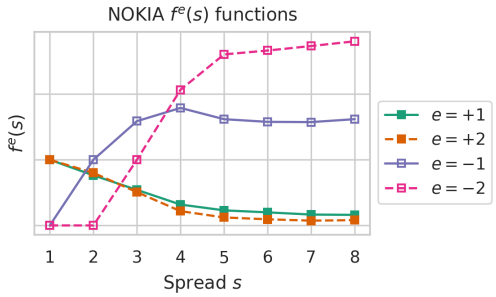


FIGURE 2 – Estimations des fonctions $\{f^e(s)\}_{e\in\mathcal{E}}$ pour NOKIA, $\mathcal{E} = \{-2, -1, +1, +2\}$.

La Figure 3 montre que les noyaux diminuent lentement selon une loi de puissance. Cette propriété de mémoire longue du spread a été observée par diverses études empiriques ; par exemple, Mike and Farmer (2008); Ponzi et al. (2006); Zawadowski et al. (2006).

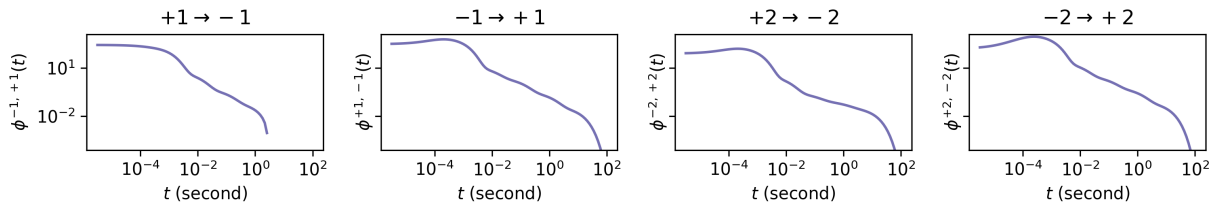


FIGURE 3 – Noyaux de Hawkes pour AXA.

Qualité de l'ajustement

Le modèle SDSH est capable de capturer les principales propriétés statistiques du processus de spread. Dans cette section de résumé, nous nous concentrons sur la présentation d'une seule propriété : la fonction d'auto-covariance (ACV) des incréments de spread (Figure 4). Des propriétés supplémentaires peuvent être trouvées dans le contenu principal de cette thèse.

La fonction d'auto-covariance normalisée des incréments de spread pendant δ secondes avec un décalage de τ secondes est définie comme suit :

$$ACV(\delta, \tau) := \frac{1}{\delta^2} Cov(S_{t+\delta} - S_t, S_{t+\delta+\tau} - S_{t+\tau})$$

La Figure 4 illustre la reproduction précise par le modèle de la courbe d'auto-covariance observée dans les vraies données. Pour des différentes valeurs de δ lorsque τ est relativement grand, toutes les courbes $ACV(\delta, \tau)$ ont tendance à converger, formant une courbe collective lisse. Cependant, lorsque τ devient trop grand par rapport à δ, la figure d'insertion montre que la courbe ACV devient bruitée. Par conséquent, pour reproduire précisément la courbe ACV sur une large gamme de τ, nous pouvons faire varier la valeur de δ et choisir des valeurs appropriées de τ ni trop petites ni trop grandes.

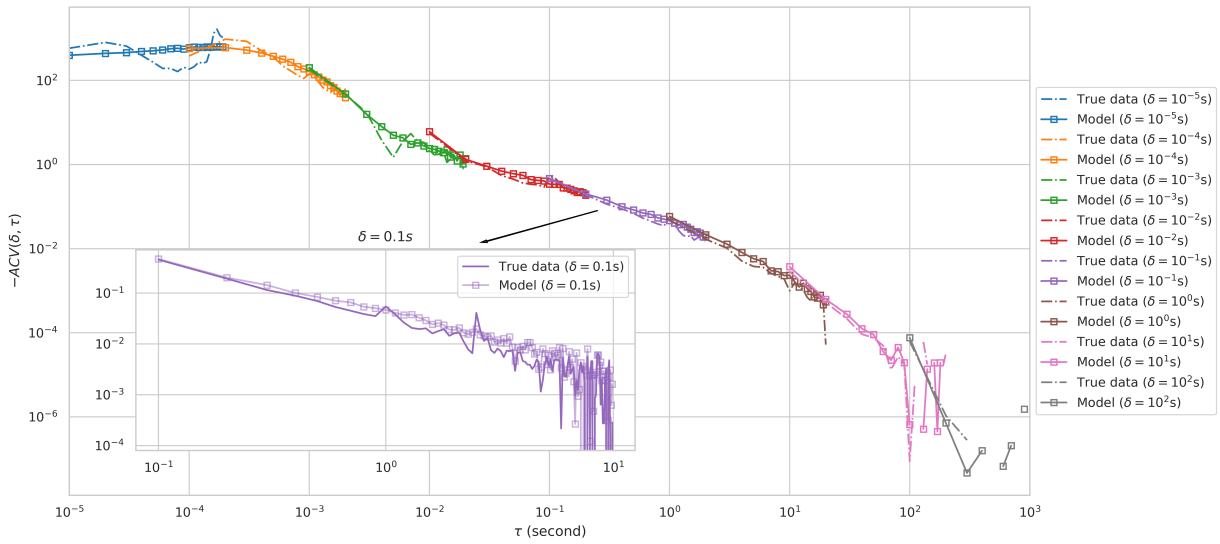


FIGURE 4 – Les fonctions $-ACV(\delta, \tau)$ pour différentes valeurs de δ en fonction de τ en utilisant une échelle logarithmique pour les vraies données AXA et les données simulées par le modèle (ajusté sur les vraies données AXA).

Prédiction

La dernière partie de ce chapitre est consacrée à la prédiction du spread, une application potentielle du modèle SDSH. Nous expérimentons avec des horizons de prédiction allant de 3 secondes à 30 secondes et comparons les capacités prédictives du modèle SDSH à la méthode ACDP introduite par Groß-KlußMann and Hautsch (2013). Les résultats comparatifs montrent que le modèle SDSH surpasse la méthode ACDP dans la plupart des cas, en particulier pour les horizons temporels courts.

II.2 Résumé du Chapitre 4

Le Chapitre 4 est un travail conjoint avec E. Bacry, T. Deschatre, M. Hoffmann et J.-F. Muzy. L'article est actuellement en préparation. Dans ce chapitre, nous étudions les processus de Hawkes avec *shot noise*, dans le but de séparer les sources endogènes et exogènes de corrélation entre les prix de deux actifs.

L'endogénéité dans le contexte financier fait référence au fait que le prix d'un actif est influencé par le prix d'autres actifs. En revanche, l'exogénéité indique les influences externes, telles que les annonces de nouvelles qui influencent les prix. Une étude récente (Marcaccioli et al., 2022) a étudié statistiquement les différentes performances des fluctuations de prix suite à des événements endogènes et exogènes. L'objectif de notre travail est de construire un modèle capable de séparer les sources endogènes et exogènes de corrélation entre les prix de deux actifs. Le Chapitre 4 se concentre principalement sur un modèle qui intègre le comportement latent des agents.

Dans un scénario simplifié, supposons que $\bar{N}_1$ et $\bar{N}_2$ sont deux processus de comptage représentant le nombre de transactions sur l'Actif 1 et l'Actif 2, respectivement. Avant de plonger dans les détails de notre modèle, examinons un modèle de processus de Hawkes classique pour $(\bar{N}_1, \bar{N}_2)$

$$\begin{aligned}
\bar{N}_1 : \lambda_1(t) &= \mu_1 + \int_0^t \varphi_{11}(t-s)\, d\bar{N}_1(s) + \int_0^t \varphi_{12}(t-s)\, d\bar{N}_2(s) \\
\bar{N}_2 : \lambda_2(t) &= \mu_2 + \int_0^t \varphi_{21}(t-s)\, d\bar{N}_1(s) + \int_0^t \varphi_{22}(t-s)\, d\bar{N}_2(s)
\end{aligned}$$

Maintenant, considérons un scénario légèrement plus compliqué. Certaines transactions sur l'Actif 1 sont déclenchées par ses propres fluctuations de prix ou par celles de l'Actif 2, reflétant une influence endogène. Pendant ce temps, certains agents détiennent les deux actifs dans leurs portefeuilles et pourraient participer à des transactions presque simultanées des deux actifs. Ces transactions fortement corrélées, connues sous le nom de comportement latent des agents, sont générées de manière exogène. Notre modèle intègre ce comportement latent des agents grâce à un processus de Poisson appelé processus de *shot noise*. Il est important de noter que le processus de *shot noise* (ou la dimension latente) est inobservable.

Dans notre modèle, $\bar{N}_1 = N_1 + N_4$ et $\bar{N}_2 = N_2 + N_5$. N_1 et N_2 sont des processus de Hawkes classiques, tandis que N_4 et N_5 sont générés par un processus de *shot noise* N_3, avec des délais (distribution exponentielle) sur les deux processus. La Figure 5 illustre le concept de notre modèle.

En référence à l'exemple 7.3(a) de Daley et al. (2003), ce modèle de processus de Hawkes avec *shot*

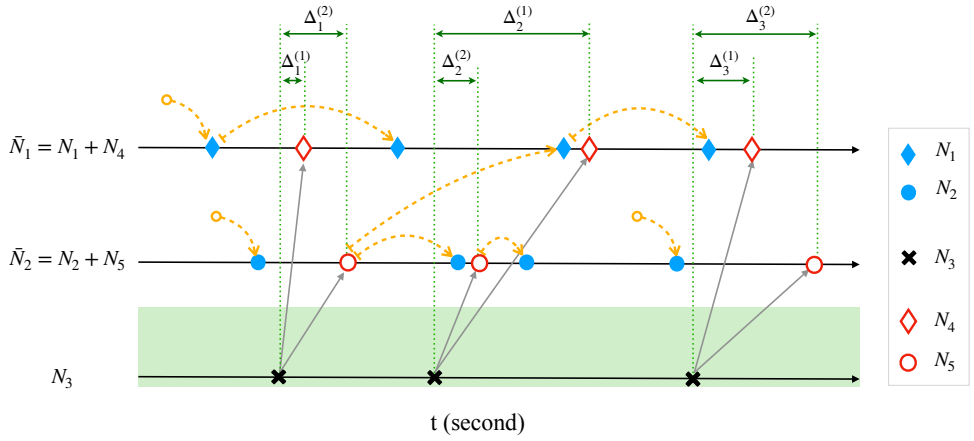


FIGURE 5 – Les flèches en pointillés jaunes montrent la relation de génération. Si une flèche pointe à partir d'un cercle vide, cela signifie que l'événement est un immigrant généré par une intensité exogène. Sinon, la flèche pointe vers un enfant à partir de son parent. Le délai du k-ième *shot noise* sur $\bar{N}_i$ est indiqué par $\Delta_k^{(i)}$ (selon notre réglage $\Delta_k^{(i)} \sim \text{Exp}(a_i)$). Le *shot noise* commun est représenté par l'ombrage vert.

noise peut être défini comme suit :

$$\begin{cases} N_1 : \lambda_1(t) = \mu_1 + \int_0^t \varphi_{11}(t-s)\, d(N_1(s) + N_4(s)) + \int_0^t \varphi_{12}(t-s)\, d(N_2(s) + N_5(s)) \\ N_2 : \lambda_2(t) = \mu_2 + \int_0^t \varphi_{22}(t-s)\, d(N_2(s) + N_5(s)) + \int_0^t \varphi_{21}(t-s)\, d(N_1(s) + N_4(s)) \\ N_3 : \lambda_3(t) = \mu_3 \\ N_4 : \lambda_4(t) = a_1 \left(N_3(t) - N_4(t)\right) \\ N_5 : \lambda_5(t) = a_2 \left(N_3(t) - N_5(t)\right) \end{cases} \tag{II.4}$$

Quelques notations

Avant de répertorier les principaux résultats de ce chapitre, clarifions d'abord certaines conventions de notation.

- $\bar{N}$ est un processus ponctuel à deux variables défini comme $\bar{N} = \begin{pmatrix} \bar{N}_1 & \bar{N}_2 \end{pmatrix}^\top$,
- φ_H est une matrice de noyaux avec $\varphi_H = \begin{pmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{pmatrix}$ et $R_H(t)$ est une matrice de fonctions définie par $R_H(t) = \sum_{n=0}^{\infty} \varphi_H^{\star n}(t)$
- Les intégrales de φ_H et R_H sont respectivement notées G_H et $\boldsymbol{R}_H$, (i.e., $G_H = ||\varphi_H||$ et $\boldsymbol{R}_H = \int_0^\infty R_H(t), dt = (I_2 - G_H)^{-1}$),
- L'intensité inconditionnelle de $\bar{N}$ est $\bar{\Lambda} = \begin{pmatrix} \bar{\Lambda}_1 \\ \bar{\Lambda}_2 \end{pmatrix} = \boldsymbol{R}_H \begin{pmatrix} \mu_1 + \mu_3 \\ \mu_2 + \mu_3 \end{pmatrix}$ et l'intensité incondition-

nelle de $\begin{pmatrix} N_1 \\ N_2 \end{pmatrix}$ est $\Lambda_H = \begin{pmatrix} \bar{\Lambda}_1 - \mu_3 \\ \bar{\Lambda}_2 - \mu_3 \end{pmatrix}$.

Théorèmes limites

En suivant les théorèmes limites dans Bacry et al. (2013b), nous pouvons prouver des versions analogues de ces résultats pour le modèle de processus de Hawkes avec *shot noise*.

Considérons l'hypothèse suivante :

$$\boxed{\begin{array}{l}\text{Pour tous les } i,j \in 1,2,\ ||\varphi_H^{ij}|| = \displaystyle\int_0^\infty \varphi_H^{ij}(t)\,dt < \infty \text{ et la matrice} \\ G_H = ||\varphi_H|| \text{ a un rayon spectral inférieur à } 1\end{array}} \tag{A}$$

Theorem. *Si la condition* (A) *est vérifiée, alors nous avons*

$$\sup_{v\in[0,1]} \left\| \frac{1}{T}\bar{N}_{Tv} - v\bar{\Lambda} \right\| \longrightarrow 0 \text{ lorsque } T \to \infty \text{ presque sûrement et en norme } L^2$$

Theorem. *Si la condition* (A) *est vérifiée, en loi pour la topologie de Skorokhod, lorsque* $T \to \infty$,

$$\frac{1}{\sqrt{T}}\left(\bar{N}_{Tv} - \mathbb{E}[\bar{N}_{Tv}]\right) \to \boldsymbol{R}_H \begin{pmatrix} \Lambda_{H,1} W_{1,v} + \mu_3 W_{3,v} \\ \Lambda_{H,2} W_{2,v} + \mu_3 W_{3,v} \end{pmatrix} \text{ for } v \in [0,1]$$

où $(W_v)_{v\in[0,1]}$ *est un mouvement brownien standard à 3 dimensions.*

Définissons $\bar{X}_t = \bar{N}_t - \mathbb{E}[\bar{N}_t]$. La matrice de covariance empirique de $\bar{N}$ sur $[0,T]$ est donnée par

$$C_{\Delta,T}(\bar{N}) = \frac{1}{T}\sum_{i=1}^{\lfloor T/\Delta \rfloor} \left(\bar{X}_{i\Delta} - \bar{X}_{(i-1)\Delta}\right)\left(\bar{X}_{i\Delta} - \bar{X}_{(i-1)\Delta}\right)^\top$$

Theorem. *Soit* $(\Delta_T)_{T>0}$ *une famille de nombres réels positifs. Supposons que* $\Delta_T/T \to 0$ *lorsque* $T \to \infty$. *Alors, nous avons*

$$C\Delta_T, T(\bar{N}) - c_{\Delta_T} \to 0 \text{ lorsque } T \to \infty \text{ en norme } L^2$$

avec

$$c_\Delta = \int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ R_H(s)\bar{\Sigma}R_H(t)^\top\,ds\,dt$$
$$+ \mu_3 \int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ \left(R_H(s) \star \bar{\Gamma}(s)\right)\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\left(R_H(t) \star \bar{\Gamma}(t)\right)^\top ds\,dt$$

où $\bar{\Sigma} = \begin{pmatrix} \bar{\Lambda}_1 & 0 \\ 0 & \bar{\Lambda}_2 \end{pmatrix}$ *et* $\bar{\Gamma}(t) = \begin{pmatrix} -a_1 e^{-a_1 t} & 0 \\ 0 & -a_2 e^{-a_2 t} \end{pmatrix}$.

Corollary (Covariance Macroscopique)**.**

$$\lim_{\Delta\to\infty} c_\Delta = \boldsymbol{R}_H \begin{pmatrix} \bar{\Lambda}_1 & \mu_3 \\ \mu_3 & \bar{\Lambda}_2 \end{pmatrix} \boldsymbol{R}_H^\top$$

Estimation (méthode NPHC)

Achab et al. (2017) ont développé une méthode d'estimation non paramétrique pour les processus de Hawkes en utilisant les trois premiers ordres des cumulants. Nous pouvons prouver que cette méthode est également applicable à notre modèle. En fait, elle est toujours efficace tant que le nombre de paramètres est inférieur au nombre d'équations de cumulants indépendantes. La Figure 6 illustre un exemple de résultats d'estimation obtenus à partir de données simulées.

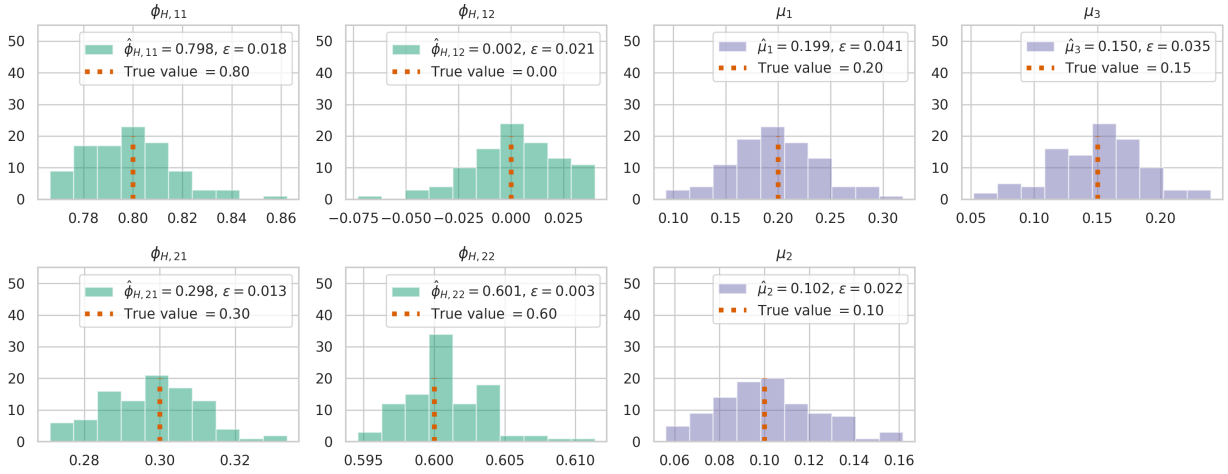


FIGURE 6 – Un exemple de normes de noyaux estimées et de *baslines* pour des données simulées. Les lignes verticales rouges en pointillés indiquent les vraies valeurs. Les histogrammes représentent les distributions des valeurs estimées à partir de 100 répétitions indépendantes. Chaque estimation est basée sur un processus simulé couvrant 10^6 secondes, soit environ $3.68 \cdot 10^6$ événements.

Extension à des dimensions supérieures

Le modèle de l'équation (II.4) peut être étendu à des dimensions supérieures. En particulier, considérons un modèle de prix bivarié, où P_1 et P_2 sont les processus de prix pour l'Actif 1 et l'Actif 2 respectivement. En suivant le modèle introduit dans Bacry et al. (2013a), la paire (P_1, P_2) peut être dérivée de $(\bar{N}_1 - \bar{N}_2, \bar{N}_3 - \bar{N}_4)$, où $\bar{N}_{1,t}$ (resp. $\bar{N}_{2,t}$) représente le nombre de sauts de prix vers le haut (resp. vers le bas) au moment t pour l'Actif 1 et $\bar{N}_{3,t}$ (resp. $\bar{N}_{4,t}$) représente le nombre de sauts de prix vers le haut (resp. vers le bas) au moment t pour l'Actif 2. Dans ce cas, les processus de comptage observables sont $(\bar{N}_i)_{i=1,2,3,4}$ et la dimension observable augmente à 4. Dans le cadre des processus de Hawkes avec modèle de *shot noise*, $\bar{N}_i = N_{H,i} + N_{D,i}$ pour $i = 1, 2, 3, 4$ où $N_{H,i}$ et $N_{D,i}$ sont définis comme suit :

$$\begin{cases} N_{H,i} : \lambda_{H,i} = \mu_{H,i} + \sum_{j=1}^{4} \int_0^t \varphi_{ij}(t-s)\, d[N_{H,j}(s) + N_{D,j}(s)] \text{ for } i \in \{1,2,3,4\} \\ N_{X,k} : \lambda_{X,k}(t) = \mu_{X,k} \text{ for } k \in \{1,2\} \\ N_{D,i} : \lambda_{D,i}(t) = a_1[N_{X,1}(t) - N_{D,i}(t)] \text{ for } i \in \{1,3\} \\ N_{D,i} : \lambda_{D,i}(t) = a_2[N_{X,2}(t) - N_{D,i}(t)] \text{ for } i \in \{2,4\} \end{cases}$$

Ici, nous supposons l'existence d'un *shot noise* à 2 dimensions, $N_{X,1}$ et $N_{X,2}$, où $N_{X,1}$ (resp. $N_{X,2}$) affecte les deux sauts de prix à la hausse (resp. les deux sauts de prix à la baisse).

La Figure 7 présente le résultat de la calibration du modèle sur les données de BNP Paribas et de Société Générale provenant d'Euronext.

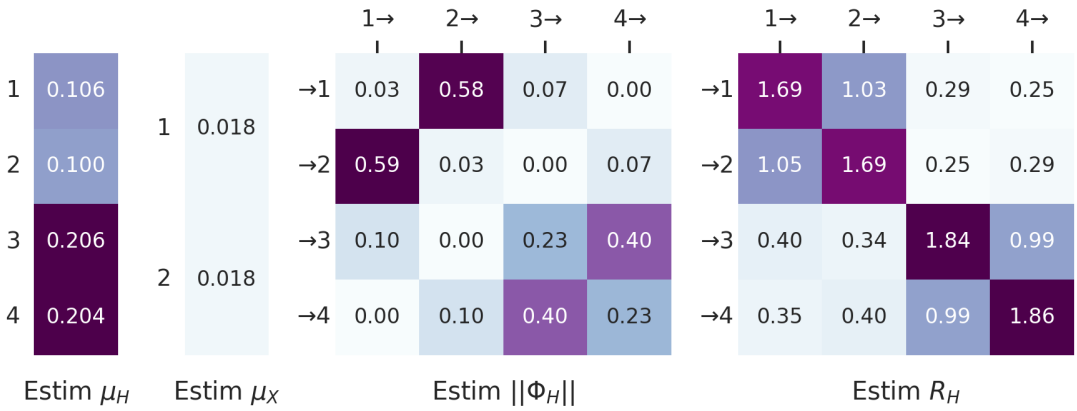


FIGURE 7 – Estimation des normes des noyaux de Hawkes et des *baselines* pour BNP Paribas et Société Générale. "1" et "2" représentent les sauts de prix à la hausse pour BNP Paribas, tandis que "3" et "4" représentent les sauts de prix à la baisse pour Société Générale.

Une variante du modèle

Les modèles précédents 1.2.4 et 1.2.5, appelés modèles de *latent-behavior shot noise*, reposent sur l'hypothèse que le processus de shot noise est le comportement latent de certains agents. Dans ce travail, nous proposons également une variante du modèle, appelée modèle de *latent-information shot noise*, où le processus de shot noise représente l'information latente du marché.

Soient $N_{H,i}(i = 1, \dots, d)$ les processus observables et $N_{X,k}(k = 1, \dots p)$ les processus latents (c'est-à-dire le processus de shot noise). L'espace des événements est $\mathcal{E} = \{N_{H,i}, i = 1, 2, \dots, d\} \cup \{N_{X,k}, k = 1, 2, \dots, p\}$ et le modèle à information latente est formulé comme suit :

$$\lambda_{H,i}(t) = \mu_{H,i} + \sum_{j=1}^{d} \int_0^t \varphi_{H,ij}(t-s)\, dN_{H,j}(s) + \sum_{k=1}^{p} \int_0^t \varphi_{X,ik}(t-s)\, dN_{X,k}(s) \text{ for } i \in \{1, 2, \dots, d\}$$

$$\lambda_{X,k}(t) = \mu_{X,k} \text{ for } k = 1, 2, \dots, p$$

où $\lambda_{H,i}$ et $\lambda_{X,k}$ sont les intensités de $N_{H,i}$ et $N_{X,k}$ respectivement. Nous pouvons prouver l'efficacité de la méthode d'estimation NPHC sur cette variante du modèle tant que $p \geqslant \frac{d^3+5d}{6(d+1)}$.

II.3 Résumé du Chapitre 5

Le Chapitre 5 est une première tentative d'utilisation des méthodes d'apprentissage profond pour classer et caractériser les agents du marché. Nos principaux objectifs sont de répondre à des questions telles que : sans s'appuyer sur des examens statistiques, pouvons-nous classifier les agents efficacement ? Comment pouvons-nous caractériser les comportements divers des agents dans le carnet d'ordres ? Quelles sont les principales caractéristiques qui distinguent différents agents ?

Dans ce chapitre, nous appliquons une méthode d'apprentissage supervisé pour classer les agents. Plus précisément, nous nous concentrons sur 28 agents très actifs sur le marché des contrats à terme

de l'indice CAC40. Nous supposons qu'à un moment donné, le comportement d'un agent peut être décrit par une séquence d'ordres qu'il a soumis. Tout au long de ce chapitre, nous constatons que l'ensemble le plus pertinent de variables descriptives de l'ordre pour cette tâche comprend trois aspects : l'ordre lui-même (heure d'arrivée, prix, taille), le contexte du marché (état du carnet d'ordres avant l'ordre) et le type d'action catégorique de l'ordre (limite, marché ou annulation à un niveau spécifique). Une entrée pour le réseau neuronal est une séquence de N ordres consécutifs passés par un agent, et la sortie est l'identifiant de l'agent.

Nous utilisons le modèle *Gated Recurrent Unit* bien connu dans la littérature, introduit par Cho et al. (2014). L'architecture du modèle pour classifier les séquences d'ordres est illustrée dans la Figure 8.

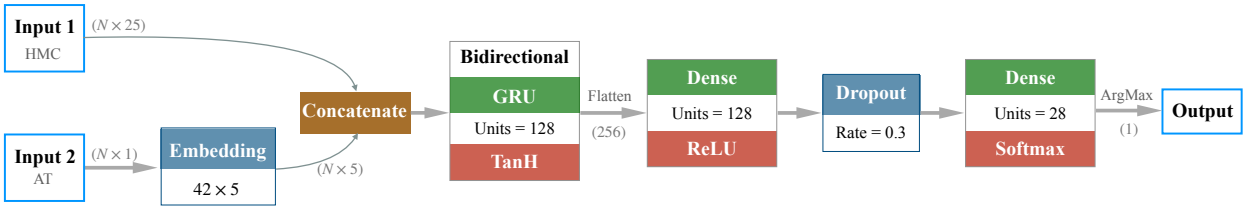


FIGURE 8 – L'Input 1 est une séquence de N ordres avec 25 caractéristiques de base HMC (l'ordre lui-même et le contexte du marché), l'Input 2 est la caractéristique catégorique AT (type d'action)

Résultats

Avec chaque entrée comprenant une séquence de 100 ordres, le modèle atteint une précision de 0,943 sur l'ensemble des données test. Ce résultat indique que les comportements des agents sont efficacement distinguables à l'aide de méthodes d'apprentissage profond.

De plus, comme illustré dans la Figure 8, la caractéristique catégorique du type d'action est représentée par un vecteur à 5 dimensions avant d'être entrée dans le GRU. Dans le succès remarquable de *Word2Vec* (Mikolov et al., 2013a,b), l'approche par *embedding* de mots capture différents degrés de similarité sémantique entre les mots (comme "homme" - "roi" = "femme" - "reine"). Inspiré par ce travail, nous visualisons les vecteurs après cet *embedding* des 42 types d'actions dans $\mathbb{R}^2$ en utilisant une projection suivant les deux axes principaux donnés par l'analyse en composantes principales. Cette visualisation révèle des motifs remarquables, comme le montre la Figure 9. Par exemple, les ordres à cours limité et les annulations sont clairement séparés, tandis que les ordres au marché se trouvent au milieu. De plus, la figure affiche un motif intéressant de la différence entre les *embedding* des ordres à cours limité et des annulations.

Expériences étendues

Dans les expériences précédentes, chaque ordre était étiqueté en fonction de son agent (ID de membre). En même temps, nous disposons également d'une autre étiquette plus fine, appelée "ITM". ITM signifie *Interactive Trading Machine* utilisée par les banques, les fonds spéculatifs et autres institutions financières. En particulier, les ITM sont une division des ID des membres, ce qui implique que chaque ID de membre peut correspondre à plusieurs ITM.

Dans les expériences ultérieures, une entrée consiste en une séquence d'ordres passés par un ITM spécifique et l'objectif est de prédire l'ID de l'ITM en tant que sortie. Tout comme dans les expériences précédentes, nous sélectionnons au total 104 ITM dans cette expérience, appartenant à 21 membres. La Figure 10 montre la matrice de confusion des résultats de classification.

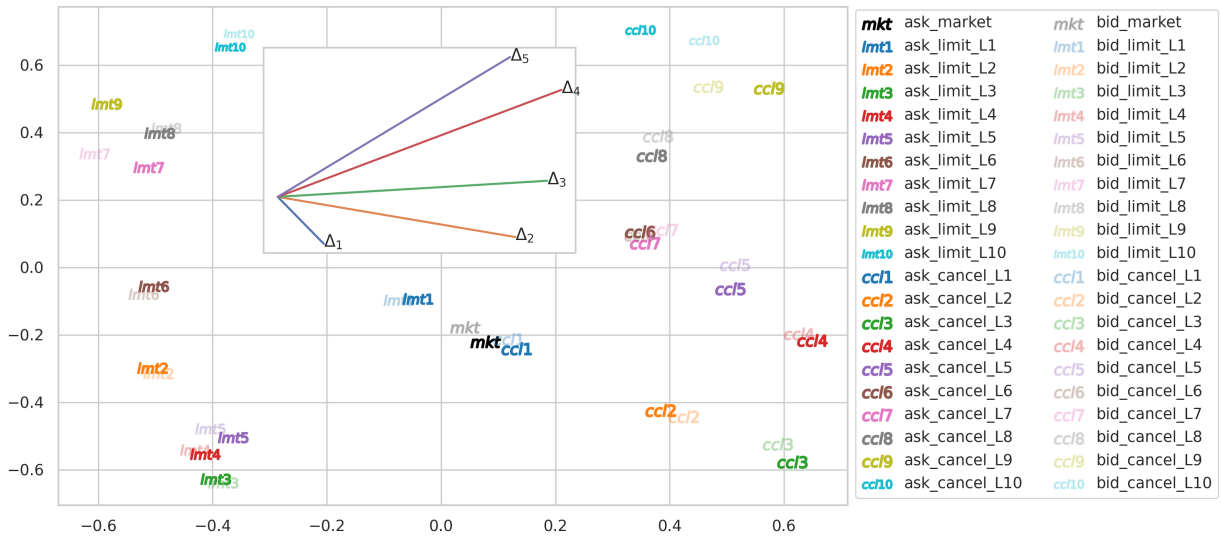


FIGURE 9 – Projection PCA à deux dimensions des *embedding* à cinq dimensions des types d'actions. La figure insérée affiche cinq lignes, chaque ligne $\Delta_i, i = 1, 2, 3, 4, 5$ indique le vecteur du niveau de cours limité i au niveau d'annulation i.

Les résultats montrent des schémas remarquables pour les ITM correspondant au même ID de membre. Par exemple, en extrayant la sous-matrice correspondant à l'ID de membre 3 et en utilisant une classification hiérarchique ascendante (clustering hiérarchique) sur les lignes de cette matrice, les 39 ITMs du Membre 3 sont répartis dans 10 sous-groupes. Des visualisations détaillées de ces résultats se trouvent dans le contenu principal de cette thèse.

II.4 Résumé du Chapitre 6

Le Chapitre 6 correspond à l'article Ruan et al. (2023a), qui a été accepté à la conférence ICAIF'23 [1]. L'objectif de ce travail est de caractériser et de regrouper les comportements des agents. Des études récentes, telles que Cont et al. (2023) et Cartea et al. (2023), ont également exploré ce sujet à travers des modèles statistiques. Cependant, dans ce chapitre, nous proposons d'utiliser une méthode d'apprentissage auto-supervisé appelée apprentissage contrastif pour développer une représentation des comportements des agents. Il convient de noter que nous ne considérons que les agents de type *market taker*.

Notre approche utilise la perte de triplet comme fonction de perte pour l'apprentissage contrastif. Introduite dans Schroff et al. (2015) pour la reconnaissance faciale, la perte de triplet vise à apprendre une représentation des entrées brutes en minimisant la distance entre les échantillons de la même classe et en maximisant la distance entre les échantillons de classes différentes. Dans notre contexte, les classes représentent différents comportements de trading de 30 agents et une entrée brute est une séquence d'ordres consécutifs de marché provenant d'un agent. Les échantillons similaires sont des séquences du même agent qui sont temporellement proches les unes des autres, tandis que les échantillons dissimilaires sont des séquences d'agents différents. Le critère de similarité est utilisé pour éviter d'imposer une seule stratégie pour chaque agent. En effet, un agent peut changer

1. The Association for Computing Machinery (ACM) International Conference on Artificial Intelligence in Finance (ICAIF) 2023

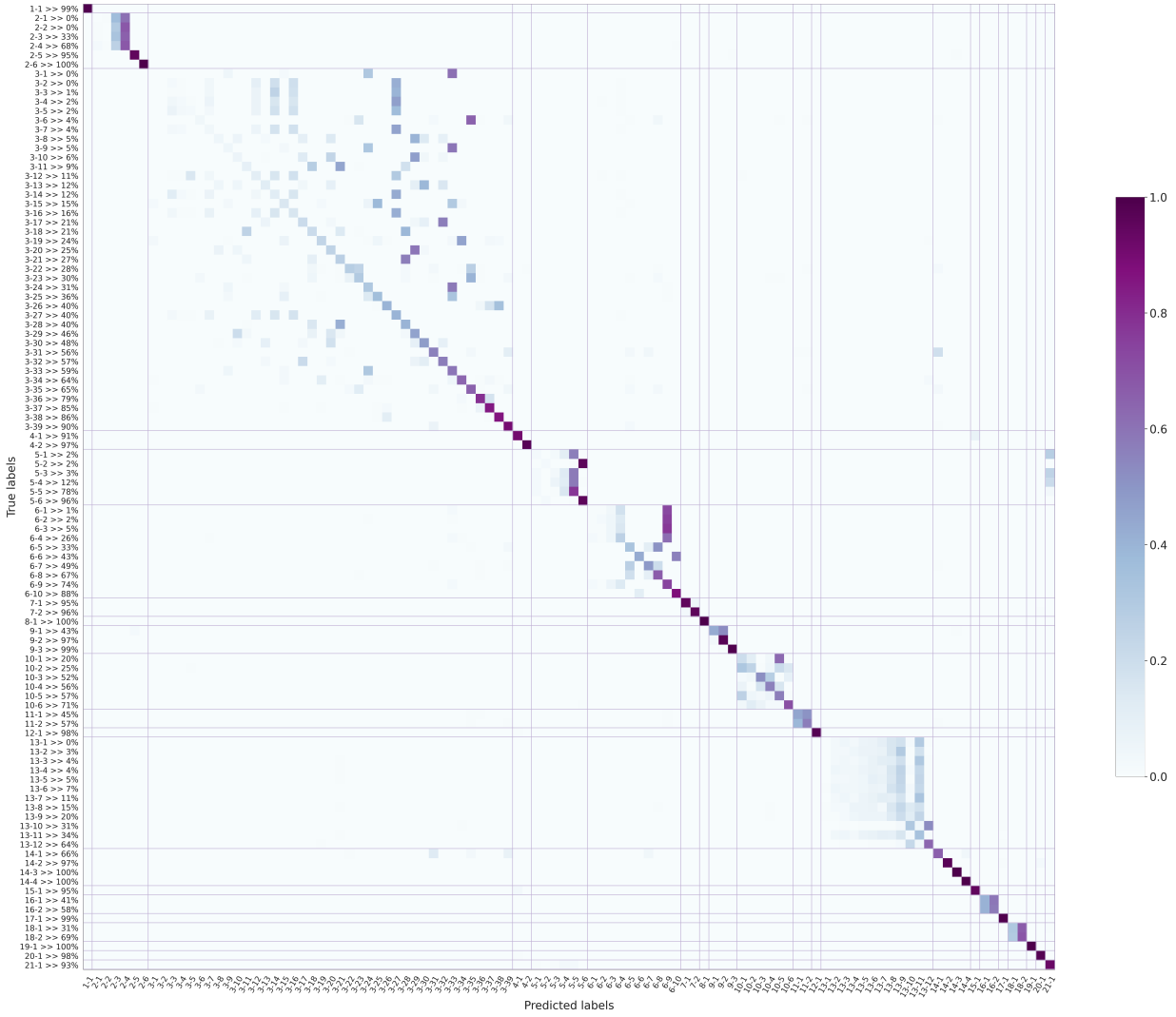


FIGURE 10 – Matrice de confusion. Les blocs de sous-matrice diagonale représentent le regroupement des ITM du même membre. Les étiquettes le long de l'axe des abscisses sont présentées au format "ITM - Membre", tandis que les étiquettes le long de l'axe des ordonnées sont affichées au format "ITM - Membre >> Précision".

de stratégie au fil du temps, en supposant que son comportement est temporairement cohérent.

La Figure 11 illustre la perte de triplet pour notre tâche. L'encodeur est un réseau à mémoire récurrente à deux couches de type LSTM (Long-Short Term Memory) (comme indiqué dans la Figure 12). Une entrée se compose d'une séquence de 50 ordres au marché (ou *market orders* en anglais), chaque ordre de marché étant caractérisé par 8 variables descriptives. Grâce à cet apprentissage contrastif, nous obtenons une fonction de représentation pour les séquences d'ordres au marché. À ce stade, la phase d'apprentissage des représentations est terminée. La phase suivante consiste en l'application de cette fonction de représentation à diverses tâches en aval.

Clustering

Nous avons choisi d'utiliser l'algorithme de clustering K-means sur les vecteurs de représentation. En appliquant cette approche, nous regroupons tous les échantillons de séquences d'ordres au

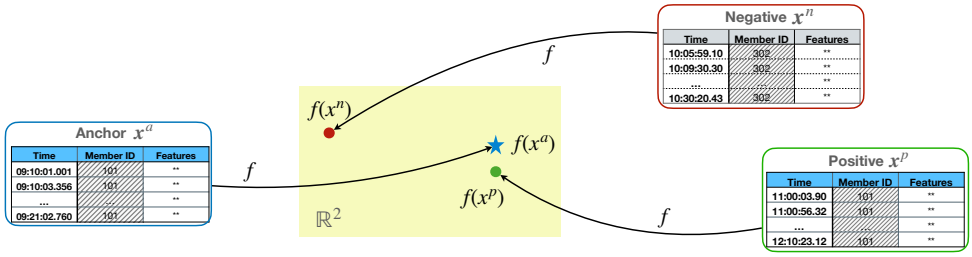


FIGURE 11 – Illustration de l'apprentissage du modèle. X est une séquence d'ordres au marché consécutifs et $f(X)$ est sa représentation vectorielle dans $\mathbb{R}^2$. La perte de triplet minimise la distance entre les échantillons du même agent $||f(X^p) - f(X^a)||_2$ et maximise la distance entre les échantillons d'agents différents $||f(X^n) - f(X^a)||_2$.

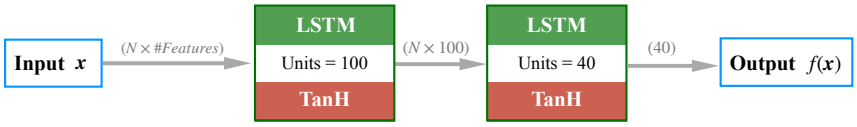


FIGURE 12 – Schéma de l'architecture du modèle d'encodage pour un échantillon.

marché en 7 groupes. La Figure 13 montre les résultats du regroupement. Notre tâche consiste maintenant à comprendre la signification de chaque groupe. Nous évaluons les groupes en fonction des indicateurs suivants : fréquence de trading, taille moyenne des transactions, spread avant les transactions, tailles de file d'attente avant les transactions, direction de trading pure accumulée et modifications de limite-à-transaction. Les résultats de l'évaluation sont résumés dans le Tableau 1.

Ces indicateurs différencient efficacement les groupes. Par exemple, le Groupe 2 présente la fréquence la plus élevée tandis que le Groupe 6 présente une faible fréquence, un spread élevé et des valeurs de direction élevées. Par conséquent, nous pouvons en déduire que le Groupe 2 correspond à un comportement de market making tandis que le Groupe 6 correspond à des pratiques de trading directionnel.

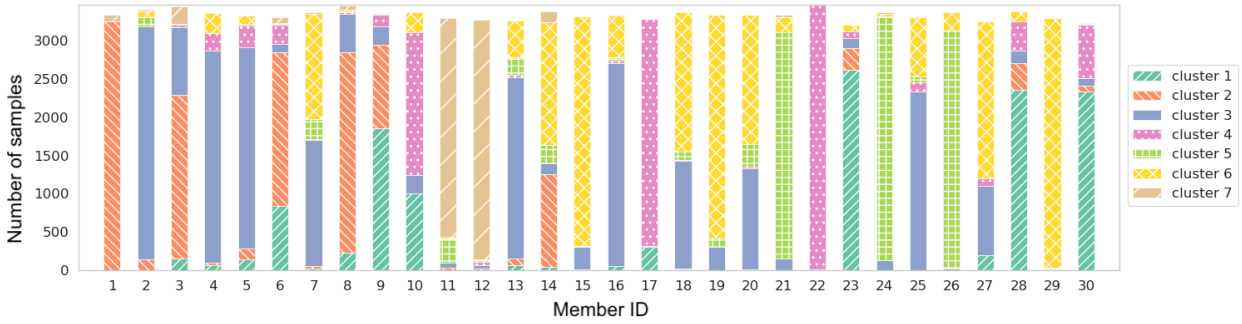


FIGURE 13 – Résultats du clustering K-means. Chaque agent est représenté par une barre verticale, qui peut être composée d'un ou de plusieurs segments. Chaque segment correspond aux échantillons de l'agent assignés à un cluster spécifique.

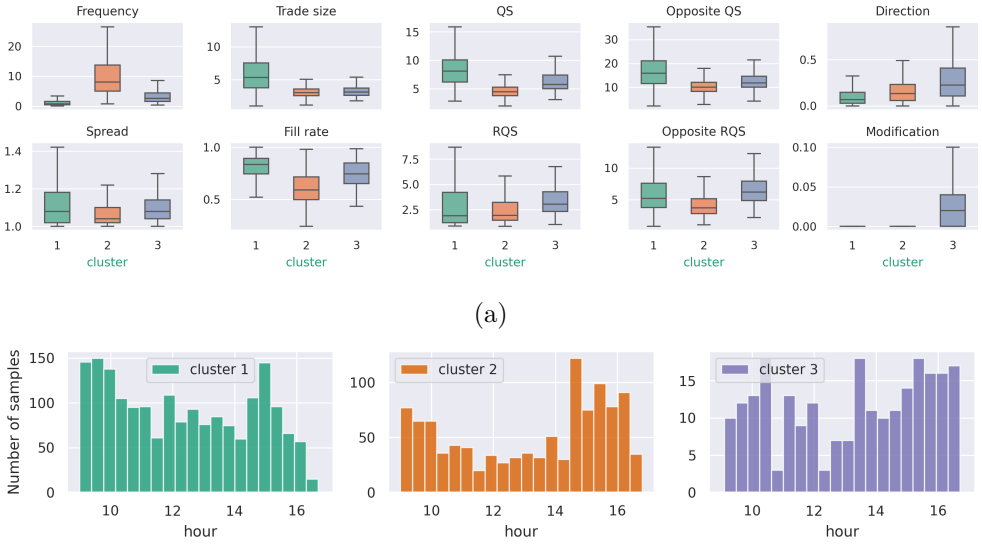
cluster	1	2	3	4	5	6	7
Frequency	+	+++	++	+	+	+	++
Trade size	+++	++	++	++	+	++	+
Fill rate	+	+	++	++	++	++	+++
Spread	++	+	++	+	+++	+++	++
QS	++	++	++	+	+++	+++	++
Opposite QS	+	+	++	++	++	++	+++
Direction	+	+	+++	+	++	+++	+
Modification			+		++	+	++

TABLE 1 – Évaluation des groupes basée sur les indicateurs ci-dessus (de none() à bas (+) à haut (+++))

Caractérisation des agents

Concentrons-nous maintenant sur un agent particulier et analysons son comportement à travers différents groupes. L'exemple que nous prenons ici est l'Agent 9, dont les ordres sont principalement classés dans les groupes 1, 2 et 3. La Figure 14 nous offre un aperçu de son comportement au sein de ces groupes. Dans le Cluster 2, l'Agent 9 s'engage dans le trading à haute fréquence, ciblant souvent les moments où la taille de la file d'attente est très faible. Lorsque nous traçons les périodes de temps de ces échantillons tout au long de la journée de trading (comme illustré dans la Figure 14b), différents comportements émergent. Nous observons qu'au matin, l'Agent 9 se comporte comme dans le Cluster 1, tandis qu'au cours de l'après-midi, il présente des comportements similaires à ceux du Cluster 2.

Prenons un autre exemple : l'Agent 10 (représenté dans la Figure 15). Nous pouvons observer que cet agent a considérablement modifié son comportement à deux reprises au cours de la période. Le premier changement survient autour de mars 2016, suivi d'un autre vers décembre 2016.



(a)

(b)

FIGURE 14 – Agent 9. (a) Chaque figure correspond à un indicateur. À l'intérieur de chaque figure, les trois barres verticales représentent les performances des échantillons de l'Agent 9 au sein de chaque cluster. (b) Chaque figure correspond à un cluster. À l'intérieur de chaque figure, l'histogramme affiche la distribution des échantillons en fonction des heures de sélection.

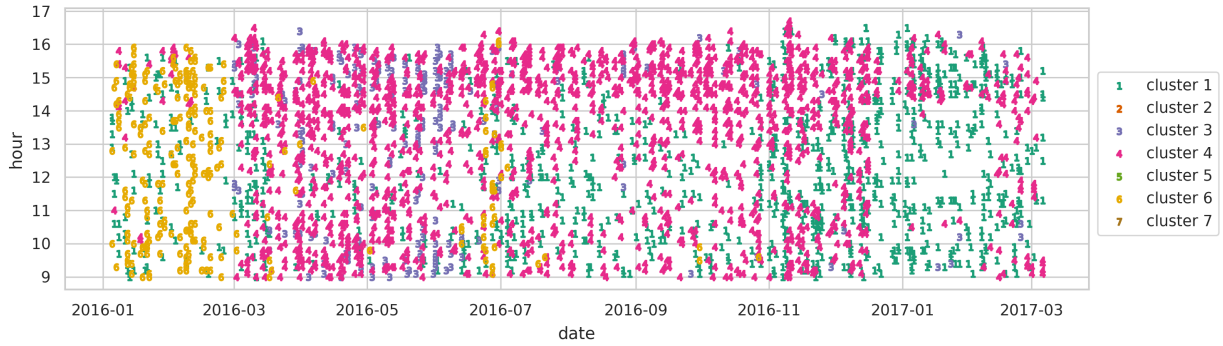


FIGURE 15 – Représentation temporelle des ordres de l'agent 10 en 2D. L'axe des abscisses représente les dates et l'axe des ordonnées représente l'heure dans une journée. Dans ce diagramme, chaque point représente l'heure d'occurrence d'un ordre de l'aget 10 et sa couleur indique le groupe auquel il appartient.

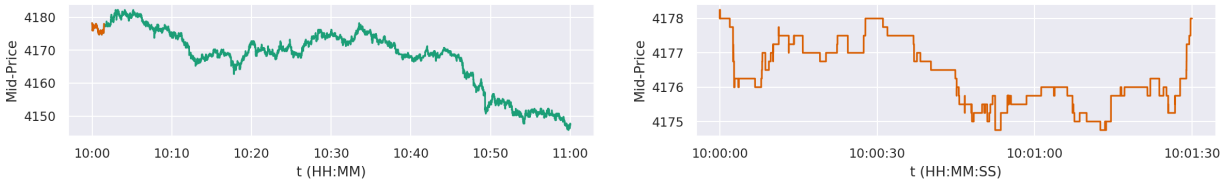
CHAPTER 1

GENERAL INTRODUCTION

1.1 - Background and Motivation

A financial market is a complex system that consists of many interacting components, including individual and institutional investors, companies, governments and regulators. For a long time, modeling the dynamics of financial markets, particularly the evolution of prices, has been a fundamental challenge for financial mathematics. The ability to accurately model and predict market behavior is crucial for developing investment strategies and managing risk.

At macroscopic time scales, the price trajectories appear continuous and resemble sample paths generated by classical models in mathematical finance, such as models driven by Brownian motion (see Figure 1.1a). However, when we zoom in to the microscopic level and examine price movements over very short time intervals (e.g., seconds), the behavior of financial processes becomes quite different. This distinct behavior is illustrated in Figure 1.1b, which zooms in on the sample process of Figure 1.1a over 90 seconds. Due to the tick structure, the price takes discrete values and manifest as jump processes. This phenomenon is called microstructure effect. While the macroscopic scale of modeling gives us a global vision of the market trend, the microscopic scale provides insight into the microstructure of the market, which is essential for understanding and describing the macroscopic behavior of the market.



(a) At the macroscopic level. The orange segment represents the initial 90-second process.

(b) At the microscopic level. The trajectory zooms in on the initial 90-second process in Figure 1.1a.

Figure 1.1 – Examples of Cac40 index Future mid-price processes

The concept of market microstructure theory emerged at the end of the 20th century, and has undergone significant evolution since 2005. This theory studies the price formation process under

specific trading mechanisms, trading information disclosure, and market components. It evaluates the efficiency and fairness of the market using typical indicators such as liquidity and price discovery. Comprehensive insights into the theoretical and empirical aspects of market microstructure can be found in books such as Bouchaud et al. (2018); Harris (2003); Hasbrouck (2007); Lehalle and Laruelle (2018); O'hara (1998). Due to the rapid development of algorithmic and electronic trading, market microstructure continues to be one of the fastest growing fields in financial research. A growing number of studies are now dedicated to the analysis and modeling of order flows and price dynamics at the microscopic level.

To describe the microstructure of order flows and price dynamics, point processes are widely used (Abergel and Jedidi, 2013; Daniels et al., 2003; Smith et al., 2003). Within this context, as reviewed in Bacry et al. (2015), Hawkes process is a very popular class of models that has proven its effectiveness in describing the dynamical properties of different quantities of limit order books (Bacry et al., 2013a; Bowser, 2007; Large, 2007; Toke, 2011). Introduced by Hawkes (1971a,b), Hawkes processes are self-exciting point processes that can capture the mutual exciting effect between events. They have gained popularity due to their ability to account for event interactions and their capacity to provide a straightforward and convincing interpretation of such interactions.

In this thesis, our objective is to examine and develop models for understanding the intricate microstructure of high-frequency data within an actual financial market. Our research is facilitated by an extensive database generously provided by Euronext Paris. This database includes a complete history of all the orders placed in the order book for 40 stocks in the CAC40 index, as well as futures contracts on the CAC40 index. In particular, each of these orders is labeled by the anonymous identification number of the initiating agent. This feature allows us to study the behavior of individual agents in the market.

More specifically, we will focus on addressing the following questions.

How can the market be resilient to liquidity shocks? (Chapter 3)

Liquidity of an asset means how easily it can be converted into cash without significantly affecting the price. It is very important in investing because it reflects the solidity of the market. Nevertheless, it is not always easy to give a accurate definition of liquidity. For a short-term view, the narrowness of the bid-ask spread can be considered as a proxy of market liquidity. When the spread is wide, it indicates a lack of liquidity and a state of disequilibrium in the market. In contrast, a narrow spread signals equilibrium. If the spread widens rapidly and substantially, it can trigger a liquidity crisis (Díaz and Escribano, 2020; Fong et al., 2017; Fosset et al., 2020a; Goyenko et al., 2009).

In **Chapter 3**, we attempt to understand how a healthy market can be resilient to shocks, in other words, how a market can "repair" itself when the bid-ask spread varies. We propose a Hawkes process model to describe the dynamics of the spread. To start, we decompose the spread processes into two increasing processes, each associated with events that amplify or diminish the spread. This decomposition transforms the spread process into a two-dimensional counting process.

Hawkes processes have demonstrated their effectiveness in modeling the mutual exciting effect between order flows, price movements and so on (Abergel and Jedidi, 2015; Bacry et al., 2013a, 2015). Nonetheless, modeling the spread by classical Hawkes processes is not straightforward due to the constraint for the spread to be strictly positive. Addressing this challenge, prior works by Zheng et al. (2014) and Fosset et al. (2020a) have explored non-linear Hawkes processes which

impose a condition that the intensity of diminishing-spread events becomes null when the spread reaches the minimum, that is one.

In this thesis, inspired by Queue reactive models (Huang et al., 2015; Wu et al., 2019), we propose a new model named the "State-dependent Spread Hawkes" (SDSH) model. This model can be considered as a generalization of the models discussed in Zheng et al. (2014) and Fosset et al. (2020a). Our SDSH model not only incorporates the memory of historical events, but also integrates the current state of the spread into the intensity function. In Chapter 3, we will demonstrate how the SDSH model enhances the model's capacity in capturing various statistical properties.

What microstructural factors contribute to correlations among the prices of different stocks? (Chapter 4)

The previous question focuses on modeling a single asset. In this question, we consider multiple assets and the correlations of their prices. Correlation is a statistical measure that determines how assets move in relation to each other. It is commonly used in finance to gain insight into the overall behavior of the larger market or to evaluate the diversification potential of a portfolio.

Assets can exhibit correlations due to a variety of factors. One of the most common sources of correlation is their sector or industry. For example, BNP Paribas and Societe Generale are both banks, so they are likely to be affected by similar macroeconomic conditions, such as interest rates, inflation, and consumer behavior. As a result, their stock prices may move in the same direction.

In addition to industry or sector affiliation, global events can also impact multiple companies and lead to correlations across sectors. Such events include press releases, government interventions, natural disasters such as Covid-19, and other geopolitical or macroeconomic shocks.

For some specific securities like futures or options, the common underlying asset of two secturities can be the most important source of their correlation. For instance, Bobl and Bund are highly correlated because they represent contracts on the same underlying asset with different maturities. (see Figure 6 in Bacry et al. (2013a))

In **Chapter 4**, we want to adress the question of the microscopic origin of the correlations. For this purpose, we introduce models that encompass different sources of correlation. The first source is the endogenous source, arising from the internal feedback mechanisms of the price processes themselves. The second source is the exogenous source, driven by the external factors, such as the news and agents behavior. The first source was studied in Bacry et al. (2013a), and the authors also established some limit theorems for the Hawkes processes model (Bacry et al., 2013b).

We propose an expanded version of the Hawkes process model introduced by Bacry et al. (2013a), which we call the "Hawkes processes with shot noise" model. Extending the classical Hawkes processes model for prices, we introduce an additional latent dimension to the model to capture the exogenous source of correlation. (In this context, "Latent" indicates that the events within this dimension are not observable). This dimension is a Poisson process and influences the prices of both assets. The rationale behind adding this latent dimension is straightforward: if the prices of two assets suddenly and simultaneously start to move significantly, a plausible explanantion is that these fluctuations are indeed driven by some common external elements.

The inclusion of the latent dimension makes the model more realistic but introduces more challenge in terms of estimation. The non-parametric cumulant method (NPHC), introduced by Achab et al. (2017), proves to be just as effective for the Hawkes shot noise model. Chapter 4 presents empirical

validation of this method's effectiveness.

What roles do market participants play in the market ? (Chapters 5 and 6)

In the two previous questions, we consider the market as a self-regulating entity. However it's important to recognize that the market is a complex system composed of numerous agents. The overall performance of the market is derived from the individual actions of each agent in the market. Consequently, understanding the roles and contributions of these agents is crucial to understanding the market as a whole.

At this point, we have to mention the Agent-base modeling (ABM) methodology that is extremely useful across various domains. It can simulate the actions and interactions of individuals and organizations in complex and realistic ways (Axtell and Farmer, 2022; Iori and Porter, 2018). However, due to privacy concerns, only a few studies (Cartea et al., 2023; Cont et al., 2023; Rambaldi et al., 2019) have been able to access the dataset with member identification. In this thesis, thanks to the access to the Euronext Paris database, which contains the agent identification, we will be able to study the trading behavior of each individual agent.

Through the examination of an agent's actions over a period, we can gain insights into some macroscopic aspects: Does the agent tend to act as a market maker or a market taker? Do they have some distinct trading strategy? How does the evolution of their strategies manifest over time? Addressing these questions is very important for both academic exploration and market regulation. These responses can be integrated into agent-based models to improve microstructural facts of the market. Additionally, they can also help regulators to identify irregular behaviors in the market.

Chapters 5 and 6 will be dedicated to addressing the questions mentioned above. We characterize an agent at a given time by a sequence of consecutive orders it executes. Chapter 5 can be viewed as a foundational step. By leveraging the supervised learning method to identify the agents, we study the importance of the features and the performance of the classification. Deep learning methods have proven their power in modeling limit order books in prior works Sirignano and Cont (2019); Sirignano (2019); Zhang et al. (2019). This chapter will further demonstrate their effectiveness in characterizing agents.

Moving to Chapter 6, we delve into addressing the earlier questions. Here, we first input agents' order sequences into a pretext task, with the aim of learning a representation of an agent's behavior at a given time. The pretext task uses a self-supervised contrastive learning approach with a triplet loss (Schroff et al., 2015). Similar to works in natural language processing, such as the remarkable success of Mikolov et al. (2013b), the learned representations can reveal some intrinsic structure within sequences of orders. Consequently, clustering algorithms can be applied to group agents with similar behaviors. We demonstrate that the agents can be clustered into groups with distinct trading strategies.

1.2 - Summary of the main results

In this section, we present a summary of the key results of this thesis.

1.2.1 Summary of Chapter 3

Chapter 3 corresponds to the paper Ruan et al. (2023b), submitted to *Market Microstructure and Liquidity (MML) Journal*. In this chapter, we propose a non-linear Hawkes process model for bid-ask spread dynamics, referred to as the "State-Dependent Spread Hawkes" (SDSH) model.

The bid-ask spread is defined as the difference between the lowest selling price and the highest buying price in a limit order book. It is usually used as a measure for market liquidity and plays a crucial role in financial analyses. In this work, we represent a spread process as $(S_t)_{t\geqslant 0}$, which can be decomposed to two terms S_t^+ and S_t^-, representing respectively the positive and negative jumps of spread. Currently, all jumps are assumed to be of size 1 tick, resulting in $S_t = S_0 + S_t^+ - S_t^-$. Before introducing our SDSH model, let us take a look at two existing models closely related to our approach. The first model is the spread model proposed by Zheng et al. (2014), which is a constrained Hawkes model with the following intensity functions:

$$\begin{aligned} \lambda_t^+ &= \mu^+ + \sum_{e\in\{+,-\}} \int_0^t \varphi^{+,e}(t-s)\, dS_s^e, \\ \lambda_t^- &= 1_{S_{t-}\geqslant 2}(\mu^- + \sum_{e\in\{+,-\}} \int_0^t \varphi^{-,e}(t-s)\, dS_s^e), \end{aligned} \tag{1.2.1}$$

where λ^+ (resp. λ^-) is the intensity associated to S^+ (resp. S^-). The indicator function $1_{S_{t-}\geqslant 2}$ ensures that the spread remains strictly positive at all times.

The second model, explored in Fosset et al. (2020a), is defined as follows:

$$\begin{aligned} \lambda_t^+ &= \mu^+ + \int_0^t \alpha\beta e^{-\beta(t-s)}\, dS_s^+ \\ \lambda_t^- &= \mu^- 1_{\{S_t\geqslant 2\}} \end{aligned} \tag{1.2.2}$$

This model is a specific instance of (1.2.1) when $\varphi^{-,+}$ and $\varphi^{-,-}$ are set to zero. The authors demonstrate that when $\alpha < 1 - \frac{\mu^+}{\mu^-}$, the Hawkes system in (1.2.2) is stable and the spread process is stationary.

The SDSH model can be seen as a extension of these two prior models. We extend (1.2.1) in two different aspects:

- The jump sizes are no longer constrained to be 1 tick. Instead, we consider the possiblity of K different jump sizes. This results in our model taking the form of a $2K$-variate Hawkes process model. The value of K is a hyperparameter that can be flexibly chosen based on the specific dataset.
- The constraint $1_{S_{t-}\geqslant 2}$ for λ^- is replaced by some more general non-negative functions $f(S_{t-})$. These new functions f, applicable to both λ^+ and λ^-, enable the model to incorporate the well-known fact that the spread is mainly mean reverting. Consequently, the term "State-dependent" is introduced. These functions f can be calibrated using available data.

In summary, we denote by S_t^e the counting process that counts the number of jumps of size e, for

$e \in \mathcal{E} := \{+1, +2, \ldots, +K, -1, -2, \ldots, -K\}$. The spread process S_t can be expressed as follows:

$$S_t = S_0 + \sum_{k=1,2,\ldots,K} kS_t^{+k} - \sum_{k=1,2,\ldots,K} kS_t^{-k}.$$

Let λ^e denote the intensity function of the counting process S^e. The SDSH model is formulated as follows:

$$\lambda_t^e = f^e(S_{t-}) \left[\mu^e + \sum_{e' \in \mathcal{E}} \int_0^t \varphi^{e,e'}(t-s) dS_s^{e'} \right]$$

Here, $f^{-k}(s)$ should be 0 when $s \leqslant k$, in order to keep the spread positive. We assume that the kernels are parameterized as a sum of L exponentials terms, given by,

$$\varphi^{e,e'}(t) = \sum_{l=1}^{L} \alpha_l^{e,e'} \beta_l e^{-\beta_l t}$$

Markov property and ergodicity

The combined process (S_t, X_t) is a Markov process, where $X_t^{e,e'} := \int_0^t \varphi^{e,e'}(t-s)\, dS_s^{e'}$.

In a simplified scenario where $K = 1$ and $L = 1$, meaning $\mathcal{E} = \{-1, 1\}$ and $\varphi_{e,e'}(t) = \alpha^{e,e'} \beta e^{-\beta t}$, we can state the following proposition:

Proposition. *The process (S_t, X_t) is **V-uniformly ergodic** under the following conditions:*

$$\boxed{\begin{array}{l} f^-(1) = 0 \\ f^-(S) \geqslant \gamma S \text{ for some } \gamma > 0 \text{ when } S \geqslant 2 \\ \sup_S \{f^+(S)\}(\alpha^{+,-} + \alpha^{+,+}) < 1 \end{array}} \tag{1.2.3}$$

The proof of this proposition can be found in 3.A.

Simulation and Estimation

By using the classical "thinning method" introduced by Lewis and Shedler (1979); Ogata (1981) and the Tick open source library (Bacry et al., 2017), the simulation is straightforward. For estimation purposes, the Maximum Likelihood Estimation (MLE) method is employed. Consider a realization on $[0, T]$ and denote by $\{t_k^e\}$ the event times on S^e. The log-likelihood function can be expressed as follows (for the sake of simplicity, we assume that $L = 1$):

$$\begin{aligned} \mathcal{L}(\alpha, \mu, f) &= \sum_{e \in \mathcal{E}} \left(- \int_0^T \lambda^e(t)\, dt + \int_0^T \log \lambda^e(t)\, dS_t^e \right) \\ &= \sum_{e \in \mathcal{E}} \sum_{k=1}^{S^e(T)} \log(\mu^e + \sum_{e' \in \mathcal{E}} \alpha^{ee'} \beta \int_0^{t_k^e} e^{-\beta(t_k^e - s)}\, dS_s^j) + \sum_{e \in \mathcal{E}} \sum_{k=1}^{S^e(T)} \log f^e(S_{t_k^e}) \\ &\quad - \sum_{e \in \mathcal{E}} \int_0^T (\mu^e + \sum_{e' \in \mathcal{E}} \alpha^{ee'} \beta \int_0^t e^{-\beta(t-s)} dS_s^{e'}) f^e(S_t)\, dt \end{aligned}$$

Here are the hyperparameters and their corresponding settings:

- K: the highest jump size allowed by the model.
- L and $\{\beta_l\}_{l=1,\dots,L}$: in practice choosing the β_l that are logarithmically spaced is sufficient to capture a wide range of behaviors, as in $\beta_l = \beta_1 10^{l-1}$.
- $f^e(s)$: we assume all $f^e(s)$ functions are constant for s exceeding a fixed value $\bar{S}$. Thus, for each e and $s \leqslant \bar{S}$, $f^e(s)$ is treated as a parameter.

An illustration of the estimation outcomes on simulated data is available in Figure 3.1 within the main body of this thesis.

Empirical results

We calibrate the SDSH model using CAC40 data from Euronext. The data corresponds to the spread processes for 3 stocks, namely AXA, BNP Paribas, Nokia, as well as the CAC40 index Future, during around 100 days. The hyperparameter settings can be found in Table 3.2.

The Figures 1.2 and 1.3 provide examples of the estimation results for f^e (Nokia) and essential kernels $\varphi^{e,e'}$ (AXA). As expected, we observe that $f^{+1}(s)$ and $f^{+2}(s)$ display a decreasing trend and approach 0 as s increases. Conversely, $f^{-1}(s)$ and $f^{-2}(s)$ are globally increasing functions. When the spread is high, small $f^{+1}(s)$ and $f^{+2}(s)$ values inhibit the positive jumps while large $f^{-1}(s)$ and $f^{-2}(s)$ values encourage the negative jumps. This mechanism effectively reinforces the mean-reversion nature of the spread.

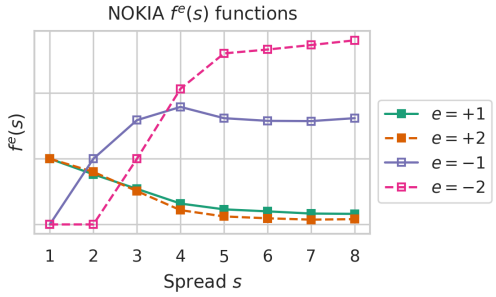


Figure 1.2 – Estimations of the $\{f^e(s)\}_{e\in\mathcal{E}}$ functions for NOKIA, $\mathcal{E} = \{-2, -1, +1, +2\}$.

Figure 1.3 shows that the kernels decrease slowly as a power-law function. This long memory property of spread has been observed by various empirical studies; for example, Mike and Farmer (2008); Ponzi et al. (2006); Zawadowski et al. (2006).

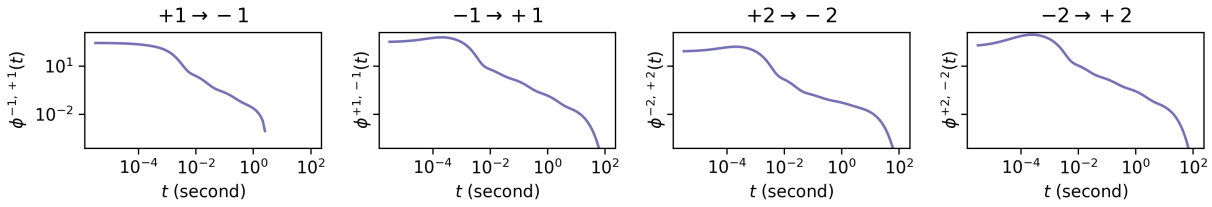


Figure 1.3 – Hawkes kernels for AXA.

Goodness-of-fit

The SDSH model is able to capture the main statistical properties of the spread process. In this summary section, we focus on presenting only one property: the auto-covariance (ACV) function of the spread increments (Figure 1.4). Additional properties can be found in the main content of this thesis.

The normalized auto-covariance function of the spread increment during δ seconds with a lag of τ seconds is defined as follows:

$$ACV(\delta, \tau) := \frac{1}{\delta^2} Cov(S_{t+\delta} - S_t, S_{t+\delta+\tau} - S_{t+\tau})$$

Figure 1.4 illustrates the model's accurate replication of the auto-covariance curve observed in the true data. For varying δ when τ is relatively large, all $ACV(\delta, \tau)$ curves tend to converge, forming a smooth collective curve. However, when τ becomes too large compared to δ, the inset figure shows that the ACV curve becomes obscured by noise. Therefore, in order to accurately reproduce the ACV curve across a broad range of τ, we can vary the value of δ and choose suitable values of τ neither too small nor too large.

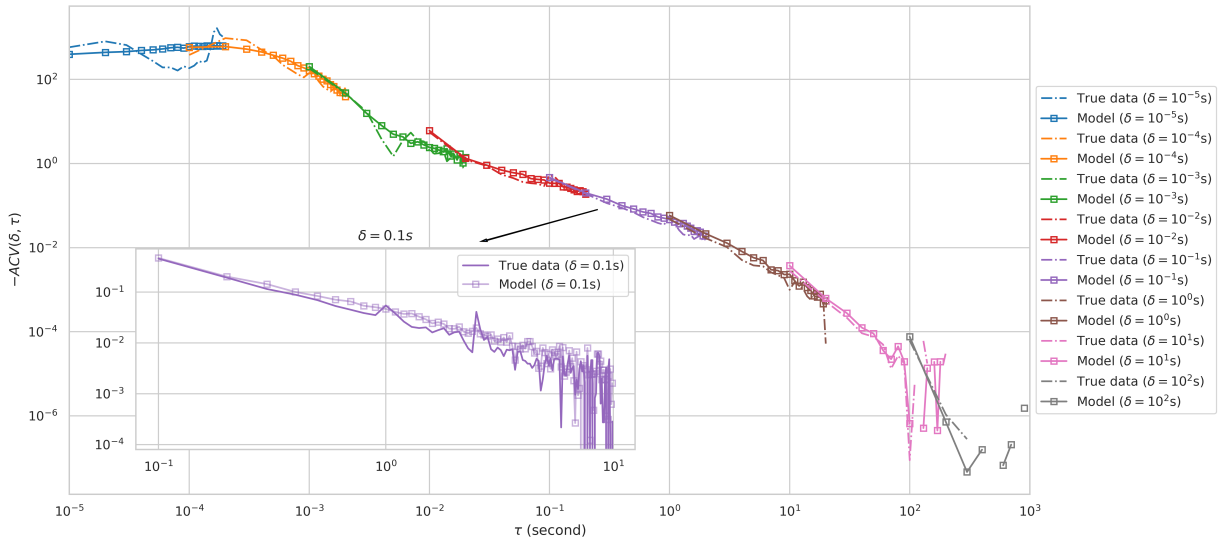


Figure 1.4 – The $-ACV(\delta, \tau)$ functions for different values of δ as a function of τ using a log-log scale both for using the AXA true data and the model-simulated data (fitted on AXA true data).

Prediction

The last part of this chapter is devoted to the prediction of spread, a potential application of the SDSH model. We experiment with prediction time horizons ranging from 3 seconds to 30 seconds and compare the predictive capabilities of the SDSH model against the ACDP method introduced by Groß-KlußMann and Hautsch (2013). The comparative results show that the SDSH model outperforms the ACDP method in most cases, especially for the short time-horizons.

1.2.2 Summary of Chapter 4

Chapter 4 is joint work with E. Bacry, T. Deschatre, M. Hoffmann and J.-F. Muzy. The paper is currently in preparation. In this chapter, we study Hawkes processes with shot noise, with the aim

of disentangling the endogenous and exogenous sources of correlation between two asset prices.

Endogeneity in the financial context refers to the fact that the price of an asset is influenced by the price of other assets. In contrast, exogeneity indicates the external influences, such as the news releases that influence prices. A recent study (Marcaccioli et al., 2022) delved into some statistical examinations of the different performances of price fluctuations following endogenous and exogenous events. The objective of our work is to construct a model which is able to disentangle the endogenous and exogenous sources of correlation between two asset prices. Chapter 4 mainly focuses on a model that incorporates the latent agent behavior.

In a simplified scenario, suppose $\bar{N}_1$ and $\bar{N}_2$ are two counting processes representing the number of trades on Asset 1 and Asset 2, respectively. Before delving into the details of our model, let us review a classical Hawkes process model for $(\bar{N}_1, \bar{N}_2)$:

$$\bar{N}_1 : \lambda_1(t) = \mu_1 + \int_0^t \varphi_{11}(t-s)\, d\bar{N}_1(s) + \int_0^t \varphi_{12}(t-s)\, d\bar{N}_2(s)$$
$$\bar{N}_2 : \lambda_2(t) = \mu_2 + \int_0^t \varphi_{21}(t-s)\, d\bar{N}_1(s) + \int_0^t \varphi_{22}(t-s)\, d\bar{N}_2(s)$$

Now let us consider a slightly more complicated scenario. Some trades on asset 1 are triggered by its own price fluctuations or by those of Asset 2, reflecting an endogenous influence. Meanwhile, there are agents holding both assets in their portfolios and they might engage in nearly simultaneous trading of both assets. These highly correlated trades, known as latent agent behavior, are generated exogenously. Our model incorporates this latent agent behavior through a Poisson process known as the shot noise process. It is important to note that the shot noise process (or the latent dimension) is unobservable.

In our model, $\bar{N}_1 = N_1 + N_4$, $\bar{N}_2 = N_2 + N_5$. N_1 and N_2 are classical Hawkes processes, while N_4 and N_5 are generated by a shot noise process N_3, with delays (exponential distribution) on both processes. Figure 1.5 illustrates the concept of our model.

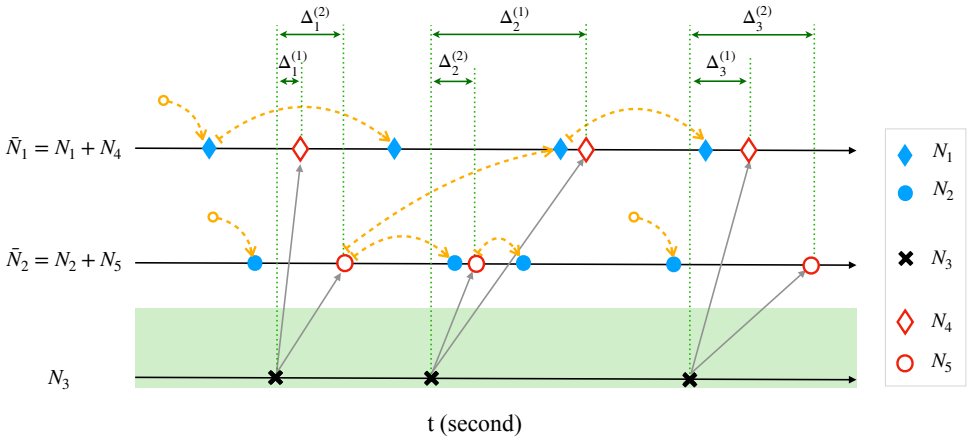


Figure 1.5 – The yellow dashed arrows show the relation of generation. If an arrow points from a empty circle, it means that the event is an immigrant generated by an exogenous intensity for self-exciting processes. Otherwise, the arrow points to a child from its parent. The delay of the k-th shot noise on $\bar{N}_i$ is indicated by $\Delta_k^{(i)}$ (by our setting $\Delta_k^{(i)} \sim \text{Exp}(a_i)$). The common shot noise is represented by the greed shade.

Referring to Example 7.3(a) in Daley et al. (2003), this Hawkes process with shot noise model can be expressed as follows:

$$\begin{cases} N_1 : \lambda_1(t) = \mu_1 + \displaystyle\int_0^t \varphi_{11}(t-s)\, d(N_1(s) + N_4(s)) + \int_0^t \varphi_{12}(t-s)\, d(N_2(s) + N_5(s)) \\ N_2 : \lambda_2(t) = \mu_2 + \displaystyle\int_0^t \varphi_{22}(t-s)\, d(N_2(s) + N_5(s)) + \int_0^t \varphi_{21}(t-s)\, d(N_1(s) + N_4(s)) \\ N_3 : \lambda_3(t) = \mu_3 \\ N_4 : \lambda_4(t) = a_1 \left(N_3(t) - N_4(t)\right) \\ N_5 : \lambda_5(t) = a_2 \left(N_3(t) - N_5(t)\right) \end{cases} \tag{1.2.4}$$

Some notation

Before listing the key results of this chapter, we first clarify some notational conventions.

- $\bar{N}$ is a two-variate point process defined as $\bar{N} = \begin{pmatrix} \bar{N}_1 & \bar{N}_2 \end{pmatrix}^\top$,
- φ_H is a kernel matrix with $\varphi_H = \begin{pmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{pmatrix}$ and $R_H(t)$ is a matrix of functions defined by $R_H(t) = \displaystyle\sum_{n=0}^{\infty} \varphi_H^{\star n}(t)$
- The integrals of φ_H and R_H are respectively denoted as G_H and $\boldsymbol{R}_H$, (i.e., $G_H = \|\varphi_H\|$ and $\boldsymbol{R}_H = \displaystyle\int_0^\infty R_H(t)\, dt = (I_2 - G_H)^{-1}$),
- The unconditional intensity of $\bar{N}$ is $\bar{\Lambda} = \begin{pmatrix} \bar{\Lambda}_1 \\ \bar{\Lambda}_2 \end{pmatrix} = \boldsymbol{R}_H \begin{pmatrix} \mu_1 + \mu_3 \\ \mu_2 + \mu_3 \end{pmatrix}$ and the unconditional intensity of $\begin{pmatrix} N_1 \\ N_2 \end{pmatrix}$ is $\Lambda_H = \begin{pmatrix} \bar{\Lambda}_1 - \mu_3 \\ \bar{\Lambda}_2 - \mu_3 \end{pmatrix}$.

Limit theorems

Following the limit theorems in Bacry et al. (2013b), we can prove analogue versions for Hawkes processes with shot noise model.

Consider the following assumption:

$$\boxed{\begin{array}{l} \text{For all } i,j \in \{1,2\},\ \|\varphi_H^{ij}\| = \displaystyle\int_0^\infty \varphi_H^{ij}(t)\, dt < \infty \text{ and the matrix} \\ G_H = \|\varphi_H\| \text{ has spectral radius smaller than 1} \end{array}} \tag{A1}$$

Theorem 1.1. *If (A1) holds, we have*

$$\sup_{v \in [0,1]} \left\| \frac{1}{T} \bar{N}_{Tv} - v\bar{\Lambda} \right\| \longrightarrow 0 \text{ as } T \to \infty \text{ almost surely and in } L^2$$

Theorem 1.2. *If (A1) holds, in law for the Skorokhod topology, as $T \to \infty$,*

$$\frac{1}{\sqrt{T}}\left(\bar{N}_{Tv} - \mathbb{E}[\bar{N}_{Tv}]\right) \to \boldsymbol{R}_H \begin{pmatrix} \Lambda_{H,1} W_{1,v} + \mu_3 W_{3,v} \\ \Lambda_{H,2} W_{2,v} + \mu_3 W_{3,v} \end{pmatrix} \text{ for } v \in [0,1]$$

where $(W_v)_{v\in[0,1]}$ is a standard 3-dimensional Brownian motion.

Set $\bar{X}_t = \bar{N}_t - \mathbb{E}[\bar{N}_t]$, the empirical covariance matrix of $\bar{N}$ on $[0,T]$ is

$$C_{\Delta,T}(\bar{N}) = \frac{1}{T}\sum_{i=1}^{\lfloor T/\Delta \rfloor} \left(\bar{X}_{i\Delta} - \bar{X}_{(i-1)\Delta}\right)\left(\bar{X}_{i\Delta} - \bar{X}_{(i-1)\Delta}\right)^\top$$

Theorem 1.3. *Let $(\Delta_T)_{T>0}$ be a family of positive real numbers. And suppose $\Delta_T/T \to 0$ as $T \to \infty$. We have*

$$C_{\Delta_T,T}(\bar{N}) - c_{\Delta_T} \to 0 \text{ as } T \to \infty \text{ in } L^2$$

with

$$c_\Delta = \int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ R_H(s)\bar{\Sigma}R_H(t)^\top \, ds\, dt$$
$$+ \mu_3 \int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ \left(R_H(s) \star \bar{\Gamma}(s)\right) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \left(R_H(t) \star \bar{\Gamma}(t)\right)^\top ds\, dt$$

where $\bar{\Sigma} = \begin{pmatrix} \bar{\Lambda}_1 & 0 \\ 0 & \bar{\Lambda}_2 \end{pmatrix}$ *and* $\bar{\Gamma}(t) = \begin{pmatrix} -a_1 e^{-a_1 t} & 0 \\ 0 & -a_2 e^{-a_2 t} \end{pmatrix}$.

Corollary 1 (Macroscopic covariance)**.**

$$\lim_{\Delta\to\infty} c_\Delta = \boldsymbol{R}_H \begin{pmatrix} \bar{\Lambda}_1 & \mu_3 \\ \mu_3 & \bar{\Lambda}_2 \end{pmatrix} \boldsymbol{R}_H^\top$$

Estimation (NPHC method)

Achab et al. (2017) developed a non-parametric estimation method for Hawkes processes by using the first three orders of cumulants. We can prove that this method is also applicable to our model. In fact, it is always effective as long as the number of parameters is lower than the number of independent cumulant equations. In Figure 1.6, an illustration of the estimation outcome on simulated data is presented. The true parameters are denoted by the red lines, while the histograms show the distributions of estimated values across 100 separate replicates. This outcome validates the efficacy of the NPHC method.

Extension to higher dimension

The model (1.2.4) can be extended to higher dimension. In particular, let us consider a bivariate price model, where P_1 and P_2 are the price processes for Asset 1 and Asset 2 respectively. Following the model introduced in Bacry et al. (2013a), the pair (P_1, P_2) can be derived from $(\bar{N}_1 - \bar{N}_2, \bar{N}_3 - \bar{N}_4)$, where $\bar{N}_{1,t}$ (resp. $\bar{N}_{2,t}$) represents the number of upward (resp. downward) price jumps at

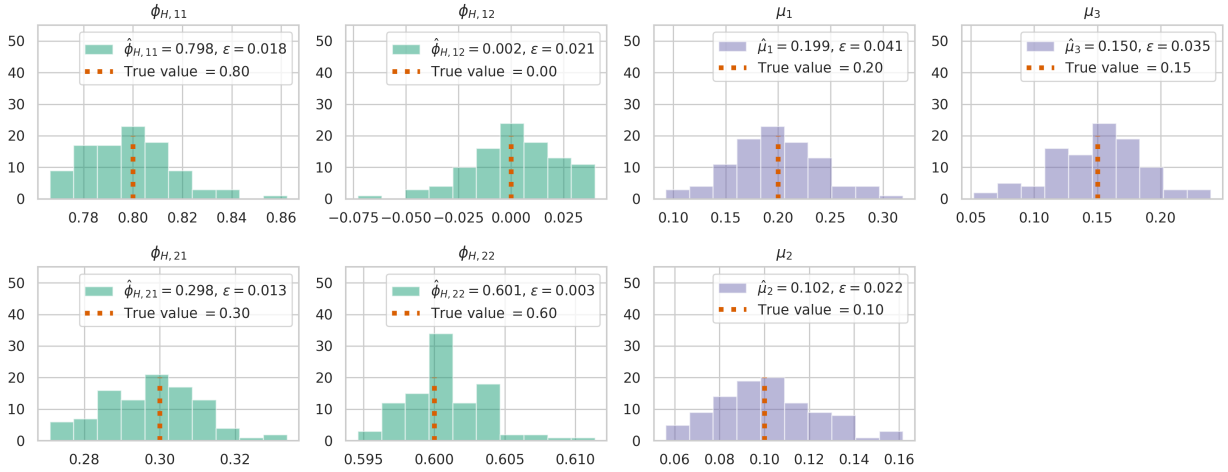


Figure 1.6 – An example of estimated kernel norms and baselines for simulated data. Red dashed vertical lines indicate the true values. The histograms represent the distributions of estimated values from 100 independent replicates. Each estimation is based on a simulated process spanning 10^6 seconds, equivalent to approximately $3.68 \cdot 10^6$ events.

time t for Asset 1 and $\bar{N}_{3,t}$ (resp. $\bar{N}_{4,t}$) represents the number of upward (resp. downward) price jumps at time t for Asset 2. In this case, the observable counting processes are $(\bar{N}_i)_{i=1,2,3,4}$ and the observable dimension increases to 4. Within the framework of the Hawkes processes with shot noise model, $\bar{N}_i = N_{H,i} + N_{D,i}$ for $i = 1, 2, 3, 4$ where $N_{H,i}$ and $N_{D,i}$ are defined as follows:

$$\begin{cases} N_{H,i} : \lambda_{H,i} = \mu_{H,i} + \sum_{j=1}^{4} \int_0^t \varphi_{H,ij}(t-s)\, d[N_{H,j}(s) + N_{D,j}(s)] \text{ for } i \in \{1,2,3,4\} \\ N_{X,k} : \lambda_{X,k}(t) = \mu_{X,k} \text{ for } k \in \{1,2\} \\ N_{D,i} : \lambda_{D,i}(t) = a_1[N_{X,1}(t) - N_{D,i}(t)] \text{ for } i \in \{1,3\} \\ N_{D,i} : \lambda_{D,i}(t) = a_2[N_{X,2}(t) - N_{D,i}(t)] \text{ for } i \in \{2,4\} \end{cases} \tag{1.2.5}$$

Here we assume the existence of a 2-dimensional shot noise $N_{X,1}$ and $N_{X,2}$ where $N_{X,1}$ (resp. $N_{X,2}$) affect the two upward price jumps (resp. the two downward price jumps).

Figure 1.7 shows the result of calibrating the model on BNP Paribas and Société Générale data from Euronext. This result offers a first blush into the model's application to real-world data. More applications will be explored in the future.

A variant of the model

The previous models 1.2.4 and 1.2.5, called the *latent-behavior* shot noise models, are based on the assumption that the shot noise process is latent behavior of some agents. In this work, we also propose a variant of the model, called the *latent-information* shot noise model, where the shot noise process stands for the latent information of the market.

Let $N_{H,i}(i = 1, \ldots, d)$ denote the observable processes and $N_{X,k}(k = 1, \ldots p)$ denote the latent processes (i.e., shot noise). The event space is $\mathcal{E} = \{N_{H,i}, i = 1, 2, \ldots, d\} \cup \{N_{X,k}, k = 1, 2, \ldots, p\}$

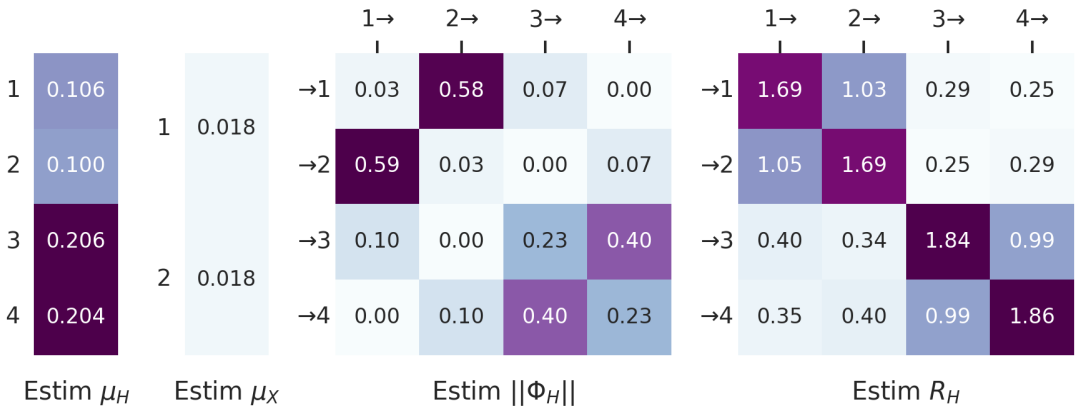


Figure 1.7 – Estimation of Hawkes kernel norms and baselines for BNP Paribas and Société Générale. "1" and "2" stand for the upward and downward price jumps for BNP Paribas while "3" and "4" stand for the upward and downward price jumps for Société Générale.

and the latent-information model is formulated as follows:

$$\lambda_{H,i}(t) = \mu_{H,i} + \sum_{j=1}^{d} \int_0^t \varphi_{H,ij}(t-s)\, dN_{H,j}(s) + \sum_{k=1}^{p} \int_0^t \varphi_{X,ik}(t-s)\, dN_{X,k}(s) \text{ for } i \in \{1, 2, \dots, d\}$$

$$\lambda_{X,k}(t) = \mu_{X,k} \text{ for } k = 1, 2, \dots, p$$

where $\lambda_{H,i}$ and $\lambda_{X,k}$ are the intensities of $N_{H,i}$ and $N_{X,k}$ respectively. We can prove the effectiveness of the NPHC estimation method on this variant model as long as $p \geqslant \frac{d^3+5d}{6(d+1)}$.

1.2.3 Summary of Chapter 5

Chapter 5 is our first attempt at using deep learning methods for classifying and characterizing agents. Our primary goals include addressing questions such as: without relying on statistical examinations, can we classify agents effectively? How can we characterize diverse agent behaviors in the limit order book? What are the main features that distinguish different agents?

In this chapter, we apply a supervised learning method to classify agents. Specifically, we focus on 28 highly active agents in the CAC40 index Future market. We assume that at a given time, an agent's behavior can be described by a sequence of orders they have submitted. Through this chapter, we find that the most relevant set of order features for this task consists of three aspects: the order itself (arriving time, price, size), the market context (state of the order book before the order) and the categorical action type of the order (limit, market or cancellation at a specific level). An input sample for the neural network is a sequence of N consecutive orders by an agent, and the output is the agent's ID.

We adopt the widely-known Gated Reccurent Unit model introduced by Cho et al. (2014). The model architecture for labelling order sequences is illustrated in Figure 1.8.

Results

With each input comprising a sequence of 100 orders, the model attains an accuracy of 0.943 on the test set. This outcome signifies that the agent behaviors are effectively distinguishable using deep learning methods.

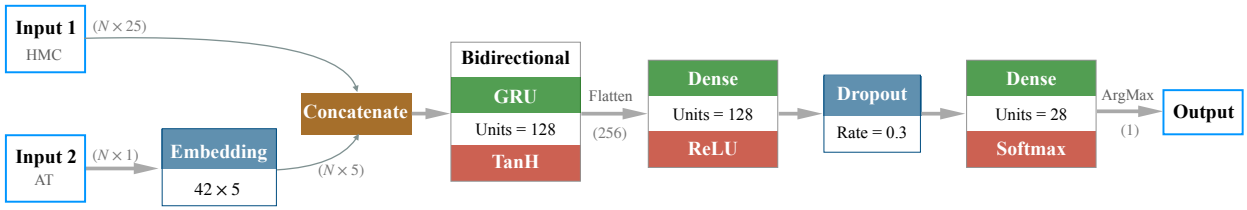


Figure 1.8 – Input 1 is a sequence of N orders with 25 basic features HMC (the order itself and the market context), input 2 is the categorical feature AT (action type)

Furthermore, as shown in Figure 1.8, the categorical action type feature is embedded into a 5-dimensional vector before being fed into the GRU. In the remarkable success of Word2Vec (Mikolov et al., 2013a,b), the word embedding approach captures various degrees of semantic simularity between words (such as "man" - "king" = "woman" - "queen"). Inspired by this work, we visualize the embedding vectors of the 42 action types in $\mathbb{R}^2$ using Principal Component Analysis projection. This visualization reveals notable patterns, as shown in Figure 1.9. For example, limit orders and cancellations are clearly seperated, while market orders are located in the middle. Moreover, the inset figure displays an interesting pattern of the difference between the embedding vectors of limit orders and cancellations.

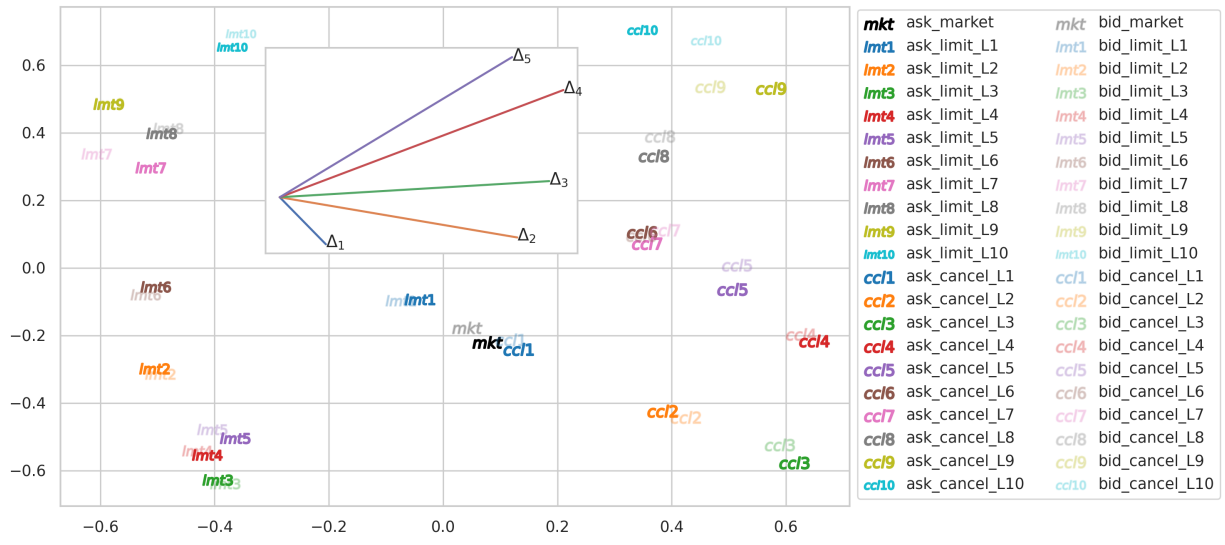


Figure 1.9 – Two-dimensional PCA projection of the 5-dimensional embedding vectors of action types. The inserted figure displays 5 lines, with each line $\Delta_i, i = 1, 2, 3, 4, 5$ indicating the vector from limit level i to cancellation level i.

Extended experiments

In the previous experiments, each order is labeled according to its agent (Member ID). At the same time, we also have another more finely-grained label, called the "ITM". ITM stands for Interactive Trading Machine used by banks, hedge funds and other financial institutions. Notably, ITMs are a division of Member IDs, implying that each Member ID can correspond to multiple ITMs.

In the subsequent experiments, an input consists of a sequence of orders placed by a specific ITM

and the objective is to predict the ITM ID as the output. Simlilar to the previous experiments, we select in total 104 ITMs in this experiment, belonging to 21 Members. Figure 1.10 shows the confusion matrix of the classification results.

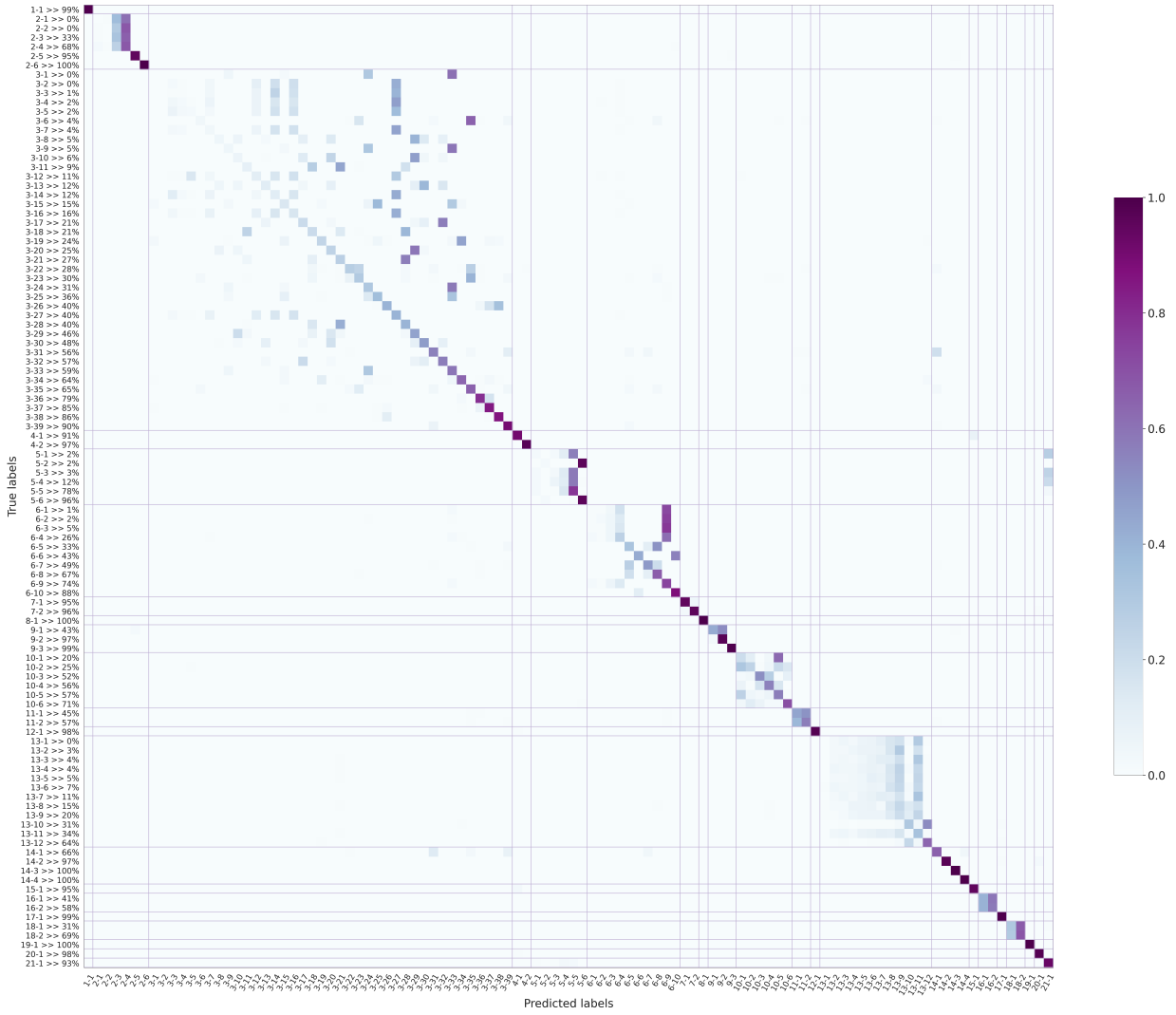


Figure 1.10 – Confusion matrix. The diagonal submatrix blocks represent the grouping of ITMs from the same Member. The labels along the x-axis (abscissa) are presented in the format "ITM - Member", while the labels along the y-axis (ordinate) are displayed in the format "ITM - Member >> Accuracy".

The results show remarkable patterns of ITMs corresponding to the same Member ID. For example, let us extract the submatrix corresponding to Member ID 3. By employing agglomerative hierarchical clustering on the rows of this matrix, we group 39 ITMs of Member 3 to 10 subgroups. Detailed visualizations of these results can be found in the main content of this thesis.

1.2.4 Summary of Chapter 6

Chapter 6 corresponds to the paper Ruan et al. (2023a), which has been accepted for publication at ICAIF'23 conference [1]. Our objective of this work is to characterize and cluster agent behaviors. Recent studys, such as Cont et al. (2023) and Cartea et al. (2023), have also explored this topic through statistical models. In this chapter, we propose to use a self-supervised learning method called contrastive learning to develop a representation of agent behaviors. It is worth noting that we only consider the liquidity-taking agents in this work.

Our approach employs the triplet loss as the loss function for contrastive learning. Introduced in Schroff et al. (2015) for face recognition, the triplet loss aims to learn a representation of raw inputs by minimizing the distance between the samples from the same class and maximizing the distance between the samples from different classes. In our context, the classes represent different trading behaviors of 30 agents and a raw input is a sequence of consecutive market orders from an agent. Similar samples are sequences from the same agent that are temporarily close to each other, while dissimilar samples are sequences from different agents. The similarity criterion is employed to avoid imposing a single strategy for each agent. In fact, an agent may change strategies over time, with the assumption that their behavior is temporally consistent.

Figure 1.11 gives an illustration of the triplet loss for our task. We aim to find an encoder f which maps the input sequences to a vector in $\mathbb{R}^n$ (in Figure 1.11, $n = 2$). The triplet loss minimizes the distance between the samples from the same agent $\|f(X^p) - f(X^a)\|_2$ and maximize the distance between the samples from different agents $\|f(X^n) - f(X^a)\|_2$. Here X^a is the anchor sample, X^p is a positive sample (i.e., similar to X^a) while X^n is a negative sample (i.e., dissimilar to X^a).

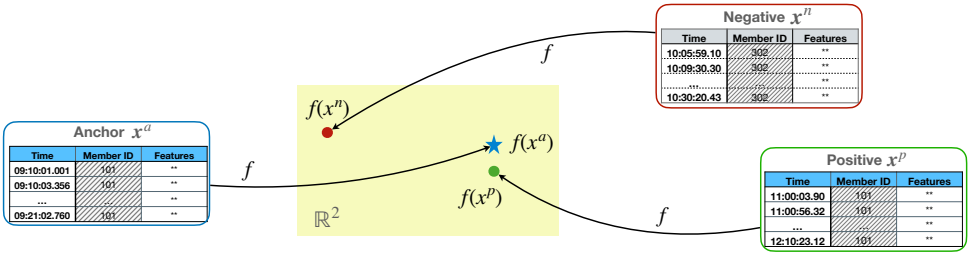


Figure 1.11 – Illustration of model learning. X is a sequence of consecutive market orders and $f(X)$ is its vector representation in $\mathbb{R}^2$.

The encoder f is a 2-layer Long-Short Term Memory (LSTM) network (as shown in Figure 1.12). An input consists of a sequence of 50 market orders with each market order characterized by 8 features. Through this contrastive learning, we get a representation function for the sequences of market orders. Until now, the preparation job is complete. The next phase involves applying the representation function to various downstream tasks.

Clustering

We choose to use the K-means clustering algorithm on the derived representation vectors. By applying this approach, we cluster all the samples of market order sequences into 7 groups. Figure 1.13 shows the clustering results. Now our task is to understand the significance of each cluster.

1. The Association for Computing Machinery (ACM) International Conference on Artificial Intelligence in Finance (ICAIF) 2023

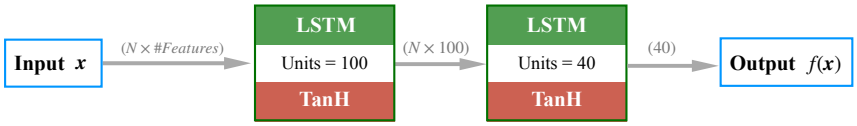


Figure 1.12 – Encoding model architecture schema for one sample.

We evaluate the clusters based on the following indicators: trading frequency, average trade size, spread before trades, queue sizes before trades, accumulated pure trading direction and limit-to-trade modifications. The evaluation results are summarized in Table 1.1.

These indicators effectively differentiate the clusters. For example, Cluster 2 demonstrates the highest frequency while Cluster 6 exhibits a low frequency, high spread and high direction values. As a result, we can derive that Cluster 2 corresponds to market making behavior while Cluster 6 corresponds to directional trading practices. More details can be found in the main content of this thesis.

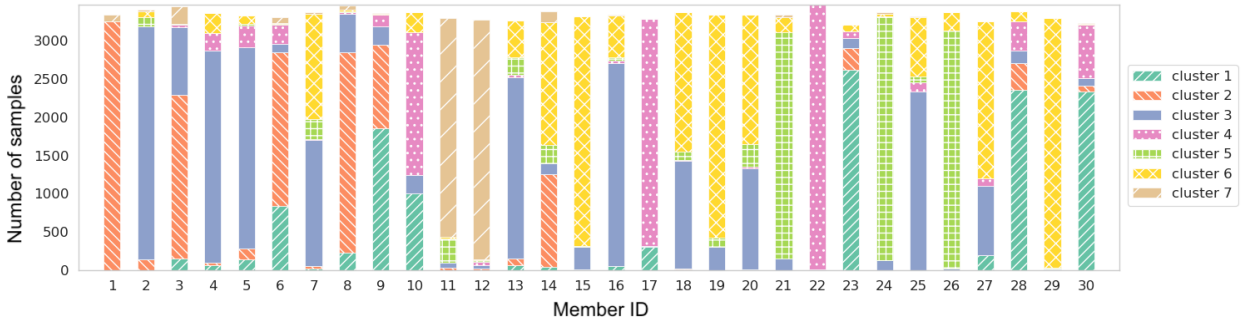


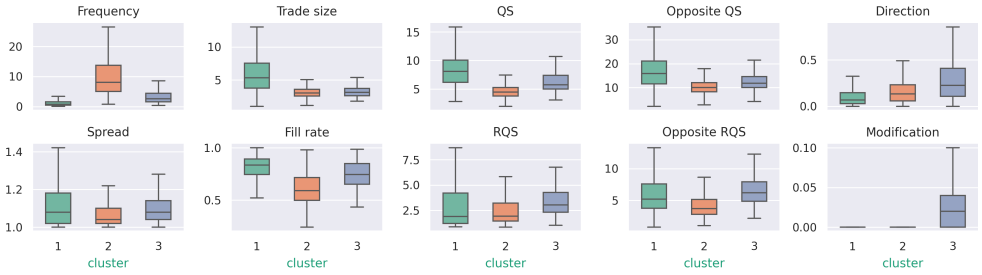
Figure 1.13 – K-means clustering results. Each agent is represented by a vertical bar, which may consist of one or multiple segments. Each segment corresponds to the agent's samples assigned to a specific cluster.

cluster	1	2	3	4	5	6	7
Frequency	+	+++	++	+	+	+	++
Trade size	+++	++	++	++	+	++	+
Fill rate	+	+	++	++	++	++	+++
Spread	++	+	++	+	+++	+++	++
QS	++	++	++	+	+++	+++	++
Opposite QS	+	+	++	++	++	++	+++
Direction	+	+	+++	+	++	+++	+
Modification			+		++	+	++

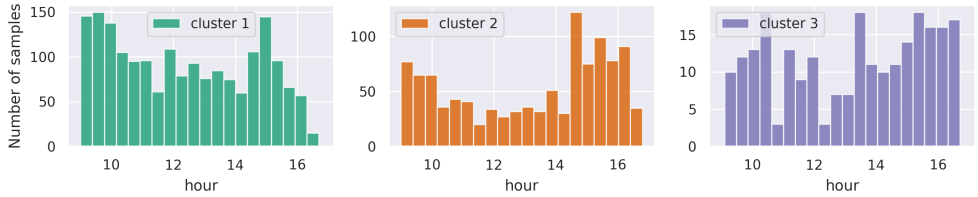
Table 1.1 – Evaluation of the clusters based on the above indicators (from none() to low (+) to high (+++)

Agent characterization

Let us now focus on a particular agent and analyze its behavior across different clusters. The example we take here is Agent 9, who is mainly assigned to Clusters 1,2 and 3. Figure 1.14 provides insight into its behavior within these clusters. When we plot the time periods of these samples throughout the trading day (as illustrated in Figure 1.14b), a distinct pattern emerges. We observe that in the morning, Agent 9 behaves as in Cluster 1, while in the afternoon, it exhibits behaviors similar to those of Cluster 2.



(a)



(b)

Figure 1.14 – Agent 9. (a) Each figure corresponds to an indicator. Within each figure, the three vertical bars represent the performance of samples from Agent 9 within each cluster. (b) Each figure corresponds to a cluster. Within each figure, the histogram plot displays the distribution of samples selecting times.

Let us consider another example, Agent 10 (depicted in Figure 1.15). We can see that Agent 10 significantly changed behavior twice during this period. The first time of change occurs around March 2016, followed by another one around December 2016.

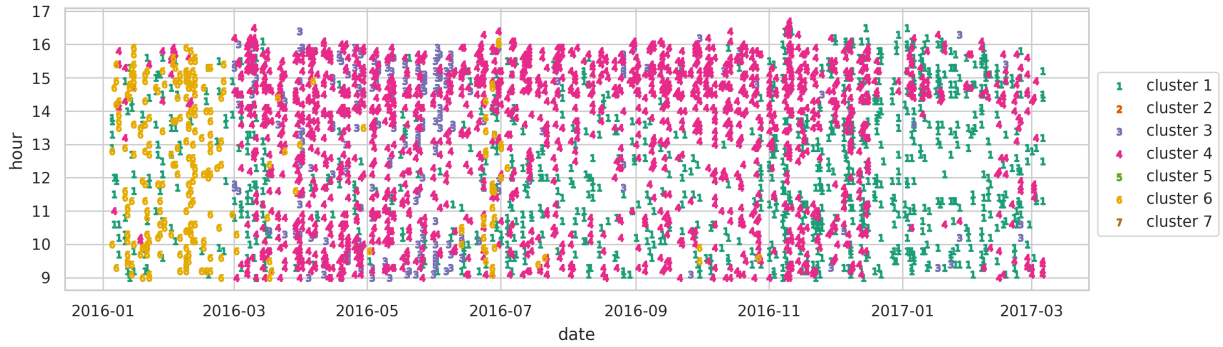


Figure 1.15 – 2-D scatter plot. X-axis represents the dates and the y-axis represents the hour in a day. In this plot, each point stands for the occurring time of a selected sample and its color shows the cluster that it belongs to.

CHAPTER 2

PRELIMINARIES

Contents

The aim of this chapter is to establish the foundational concepts and notations that will be used throughout the thesis. Section 2.1 provides an overview of the financial market, as well as an introduction to the dataset used in this thesis. In section 2.2, we will introduce the Hawkes processes and some of their properties. The simulation and estimation inferences are also discussed in this section. These preparatory elements lay the groundwork for Chapters 3 and 4. Section 2.3 will introduce some most popular artificial neural networks and their variants, setting the stage for Chapters 5 and 6.

2.1 - Financial market

Financial markets refer to various markets where the buying and selling of securities takes place, such as stock market, bond market, forex market, and derivatives market. The primary objective

of financial markets is to facilitate the efficient allocation of capital and assets in a financial economy. Over the past few decades, financial market have evolved from traditional markets, which were manual and labor-intensive with high transaction costs, to modern financial markets. The modern market refers to the digitalisation of financial markets and the rise of new financial technologies, including high-frequency trading (HFT) and algorithmic trading. These advancements enable traders and investors to buy and sell securities quickly and efficiently.

This section is mainly dedicated to the introduction to some basic concepts of financial market and description of the data used in this thesis.

2.1.1 High-Frequency Trading

High-frequency trading (HFT) refers to the use of advanced algorithms to make trades at lightning-fast speeds. HFT is largely employed by major investment banks, hedge funds and institutional investors. Its evolution began gradually after NASDAQ introduced a purely electronic form of trading in 1983, and the execution time diminished from several seconds to milliseconds and even microseconds. The key characteristics are trading at high speed, a large number of transactions and short-term investment horizons. All portfolio-allocation decisions are executed by computerized quantitative models. According to Biais et al. (2014), HFT strategies can be classified into five types: market-making, arbitrage, directional trading, structural trading, and manipulation.

2.1.2 Limit Order Book

A market refers to a place where there are both buyers and sellers. Buyers always aim to buy at lower price while sellers strive to sell at higher price. In financial market, the limit order book provides such a mechanism for facilitating the interactions between buyers and sellers.

A **Limit Order Book** (LOB) is an electronic continuous-time double-auction mechanism which records of all active orders for a particular financial asset at a given time Gould et al. (2013). It displays the collection of buy and sell intentions within a specific market and comprises of two sides: the bid side and the ask side. The bid side consists of the available prices of the buy orders, while the ask side consists of the available prices of the sell orders. These prices are discrete and have a basic unit of price interval referred to as the **tick size**, which represents the smallest increment in price for the security.

The **Best Bid Price** (or simply the "best bid") is the highest buying price among the active buy orders at time t. This price is also often referred to as the first level of the bid side. In a similar manner, the **Best Ask Price** is defined as the lowest selling price among the active sell orders. The best bid and best ask prices are denoted as $P_1^b(t)$ and $P_1^a(t)$ respectively. Furthermore, the kth level of the bid (or ask) side refers to the kth highest (or lowest) available price in the bid (or ask) side. In this thesis, $P_k^b(t)$ and $P_k^b(t)$ are used to denote the bid and ask prices of the kth level respectively.

Simultaneously, the volume of available limit orders at the kth level of the bid side is represented by $V_k^b(t)$, while the volume of available limit orders at the kth level of the ask side is denoted by $V_k^a(t)$.

An order $\boldsymbol{x}$ can be characterized by several features including its arrival time t_x, side (bid or ask) s_x, price P_x, size $V_x > 0$, and order type C_x. These features can be represented in vector form as $\boldsymbol{x} = (t_x, s_x, P_x, V_x, C_x)$.

In the following, we will show the three most common types of orders (C_x): limit orders (Type=1), market orders (Type=2), and cancellation orders (Type=3).

- **Limit order :** An order $\boldsymbol{x}$ is a bid limit order if $s_x = Bid$ and $P_x < P_1^a(t_x-)$. $\boldsymbol{x}$ is an ask limit order if $s_x = Ask$ and $P_x > P_1^b(t_x-)$. (Order type = 1)
- **Market order :** An order $\boldsymbol{x}$ is a bid market order if $s_x = Bid$ and $P_x \geqslant P_1^a(t_x-)$. $\boldsymbol{x}$ is an ask limit order if $s_x = Ask$ and $P_x \leqslant P_1^b(t_x-)$. A market order is an order to buy or sell immediately. (Order type = 2)
- **Cancellation order :** A limit order which has not been filled can be removed (fully or partially), this action corresponds to a cancellation order. (Order type = 3)

Now let us take the figure as an example to better understand some key terms in an LOB:

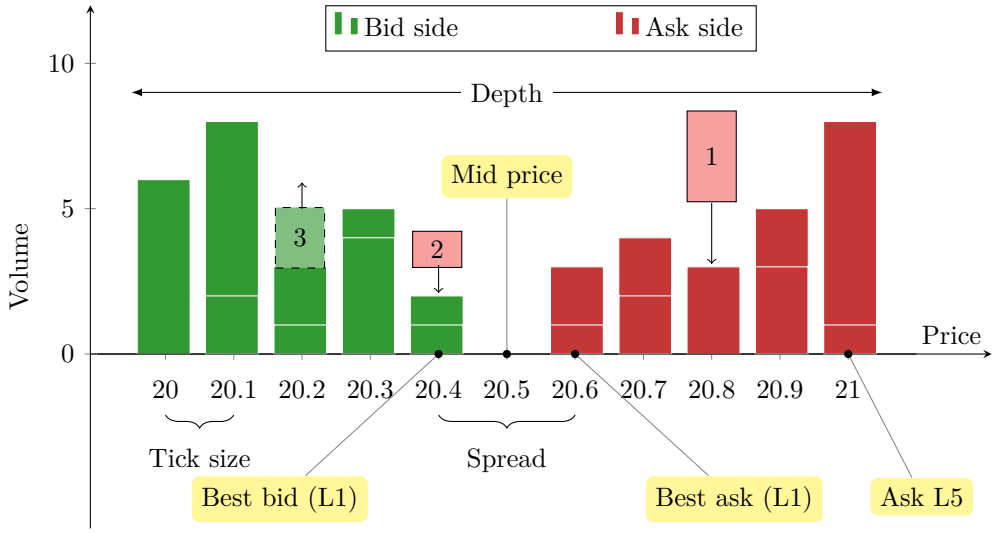


Figure 2.1 – A snapshot of an LOB at a point of time

- **Tick size** : The tick size of an LOB is the smallest permissible price interval (in the example Fig. 2.1, tick size is 0.1)
- **Best ask** : The best ask at time t of an LOB is the lowest ask price (20.6 in the Fig. 2.1)
- **Best bid** : The best bid at time t of an LOB is the highest bid price (20.4 in the Fig. 2.1)
- **Mid-price** : The mid-price at time t of an LOB is the average of the best bid and best ask prices at time t (20.5 in the Fig. 2.1)
- **Spread** : The spread at time t of an LOB is the difference between the best ask price and the best bid price (in the example figure Fig. 2.1, the spread is 0.2, i.e. 2 ticks)

Abergel et al. (2016); Bouchaud et al. (2018); Gould et al. (2013) provide a nice introduction to the LOB and its properties.

2.1.3 Market participants

There are various participants in a financial market, we will list some of them here.

Market makers

Market makers, also known as liquidity providers, play a crucial role in the functioning of a market. They are typically large banks or financial institutions. They post liquidity on both sides of the

market (buy and sell) and attempt to earn the bid-ask spread. By doing so, they take an adverse selection risk at the same time, if the prices is really changing, they will never buy-back at a good price.
Some market makers have a mandate to always provide liquidity to the market, while others may only do so in certain market conditions. During periods of high volatility or uncertainty, some market makers may withdraw from the market, leading to a temporary reduction in liquidity. In fact, many market makers also engage in short-term trading strategies and may act as market takers. This means that they will sometimes take positions in the market based on their own analysis and research, rather than simply providing liquidity.

Institutions

Institutions are large entities such as banks, which engage in significant investments and thus gain importance within the market. For instance, Proprietary trading firms and Investment banks are two types of institutions.

A proprietary trading firm, refers to a financial institutions that trade for their own account, using the firm's capital rather than client funds. The firm may maintain the full amount of gain earned on the investment by doing so.
An investment bank functions as a financial services company that acts as an intermediary in large and complex financial transactions. Notable global investment banks include JPMorgan Chase, Goldman Sachs and others. They are usually involved in managing complex financial transactions like IPOs and mergers for corporate clients.

Hedge funds

A hedge fund is a limited partnership of private investors whose money is managed by professional fund managers who use a wide range of strategies. The term "hedge" originally referred to the practice of using various techniques to mitigate or "hedge" against market risks, but modern hedge funds often use a broader array of strategies. Common hedge fund strategies includes long/short equity, arbitrage, global macro, event-driven and so on.

Brokers

For many individual investors, direct access to the securities markets can be difficult without the assistance of a broker. A broker is an individual or firm that acts as an intermediary between an investor and a securities exchange. Brokers make their money by charging their customers a commission for their services. They have a main role in facilitating the trading of securities on behalf of their clients, which involves executing orders to buy or sell securities in the markets.

2.1.4 Data presentation

Within this thesis, most of the numerical experiments rely on a database sourced from the french Euronext market's CAC 40. This database includes a complete record of the limit order book for 40 stocks within the CAC40 index, as well as the futures contract associated with the CAC 40 Index. The database, provided by Euronext, covers the period from February 2017 to February 2018 for the stocks and from January 2016 to August 2017 for the futures contract. The data records every single modification in the limit order book, across 15 limit levels on each side. The timestamps have a precision of 1 microsecond.

CAC 40

CAC 40 is a stock index, which is composed of the 40 most traded stocks on Euronext Paris. It is a so-called "capitalization-weighted" index, which means that the market capitalization (number of shares in circulation multiplied by the current share price) determines the weight of each company in the index.

CAC 40 index Future (FCE)

A futures contract is an agreement between two parties to exchange a certain quantity of an underlying asset at a predetermined price at a specified time in the future. Underlying assets include physical commodities or other financial instruments. Normally, futures contracts are traded on an exchange and to facilitate trading, the exchange specifies standardized features of the contract.

As its name indicates, the Future CAC 40 (CAC 40 Index Future; FCE) has as its underlying value the Paris stock market index, the CAC 40.

Data example

Table 2.1 provides an artificial example of several consecutive actions on the limit order book. It is worth noting that market context columns are visible to traders and investors, whereas the order columns can only be viewed by the exchange. We are grateful to Euronext for providing us with access to the order book data of the CAC 40 index. This access allows us to observe the real-time evolution of the limit order book, including the order features (agent ID, order side, etc).

Time	Order $\boldsymbol{x}$					Market Context before the order $\boldsymbol{x}$											
	ID	Side	Order type	Price	Size	Mid	Bid side						Ask side					
							Price			Size			Price			Size		
t_x	A_x	s_x	C_x	P_x	V_x	P^m	P_1^b	P_2^b	P_3^b	V_1^b	V_2^b	V_3^b	P_1^a	P_2^a	P_3^a	V_1^a	V_2^a	V_3^a
09:54:22.297	1	Ask	1	20.8	3	20.5	20.4	20.3	20.2	2	5	5	20.6	20.7	20.8	3	4	3
09:54:22.509	2	Ask	2	20.4	1	20.5	20.4	20.3	20.2	2	5	5	20.6	20.7	20.8	3	4	6
09:54:23.655	1	Bid	3	20.2	2	20.5	20.4	20.3	20.2	1	5	5	20.6	20.7	20.8	3	4	6
09:54:23.985	3	Ask	2	20.4	3	20.5	20.4	20.3	20.2	1	5	3	20.6	20.7	20.8	3	4	6
09:54:24.003	4	Ask	1	20.5	2	20.35	20.3	20.2	20.1	1	5	3	20.4	20.5	20.6	2	0	4

Table 2.1 – An artificial example of 5 consecutive actions on the limit order book (only 3 limit levels are shown in this table and the timestamp precision is one millisecond in this example). For example, the first order in the table represents the following scenario: At 9:54:22.297, Agent 1 added a limit order (order type = 1) of size 3 at level 3 of the ask side. As a result, the order volume at level 3 of the ask side increased from 3 to 6 after this order from Agent 1

2.2 - Hawkes process

Hawkes process, which was first introduced by Alan G. Hawkes in Hawkes (1971a,b), is a type of self-exciting point process. Originally developed to model earthquake events in seismology (Hawkes, 1973), Hawkes process has since been widely studied and applied to various fields, ranging from epidemiology and social networks to neuronscience and finance. In epidemiology, Hawkes process has been used to model the spread of infectious diseases (Rizoiu et al., 2018). For instance, many recent studies involve in modelling the dynamics of Covid-19 pandemic using Hawkes processes (Chiang et al., 2022; Garetto et al., 2021). It is also applied in modelling social networks, (Kobayashi and Lambiotte, 2016; Rizoiu et al., 2017) and in neuronscience (Lambert et al., 2018; Reynaud-Bouret et al., 2013). In addition to these applications, Hawkes process is particularly popular and extensively studied in finance. It is usually used to model financial market, such as the arrival of trades and order flows of limit order books (Bacry et al., 2013a, 2016; Bacry and Muzy, 2014; Morariu-Patrichi and Pakkanen, 2022; Rambaldi et al., 2019; Wu et al., 2019). In particular, Bacry et al. (2015) provides a comprehensive review of its application in finance. A recent book (Laub et al., 2021) gives an overview of the crucial aspects of Hawkes processes and their applications.

2.2.1 Point processes

Before proceeding to the details of Hawkes process, let us first introduce some preliminary definitions.

Definition 2.1 (Counting Process)**.** *A d-dimensional* ***counting process*** $\{\boldsymbol{N}_t, t \geqslant 0\}$ *is a stochastic process which satisfies the following three properties for all* $i \in \{1, 2, \dots, d\}$*:* $\boldsymbol{N}_{i,t} \geqslant 0$*,* $\boldsymbol{N}_{i,t} \in \mathbb{N}$ *and if* $s \leqslant t$*,* $\boldsymbol{N}_{i,s} \leqslant \boldsymbol{N}_{i,t}$*.*

We use $\mathcal{F}_t$ *to represent the filtration for the history up to time* t*.*

Definition 2.2 (Point Process)**.** *A d-dimensional* ***point process*** *is a sequence of random event arrival times* $\boldsymbol{T} = \{(t_0, e_0), (t_1, e_1), (t_2, e_2), ...\}$*, for all* $k \in \mathbb{N}$*.* t_k *stands for the arrival timestamps and* $e_k \in \{1, 2, \dots, d\}$ *stands for the event type. For all* $k \in \mathbb{N}$*,* $t_k > 0$ *and* $t_k < t_{k+1}$*.*

The terminology of point process and counting process is usually interchangeable. From a point process $\boldsymbol{T}$, we can define a right-continuous counting process $\{\boldsymbol{N}_t\}$ by $\boldsymbol{N}_{i,t} = \sum_{k \geqslant 0} \mathbb{1}_{\{t_k \leqslant t\} \cap \{e_k = i\}}$, in this case $\boldsymbol{N}_t$ is called the counting process associated with the point process $\boldsymbol{T}$. In this thesis, we use both the two notations to represent a point process. In this thesis, we assume that the counting and point processes are simple, meaning that each event arrival results in a jump of exactly 1.

Definition 2.3 (Conditional intensity)**.** *The intensity of a counting process is a measure of the rate at which events occur. If* $\{\boldsymbol{N}_t, t \geqslant 0\}$ *is a d-dimensional counting process with associated filtration* $\mathcal{F}$*, then its intensity vector at time* t *given the past (before* t *but excluding* t*) is* $\boldsymbol{\lambda}_t = (\lambda_{1,t}, \lambda_{2,t}, ..., \lambda_{d,t})$ *where*

$$\lambda_{i,t} = \lim_{h \searrow 0} \frac{1}{h} \mathbb{E}[N_{i,t+h} - N_{i,t} \,|\, \mathcal{F}_{t-}]$$

2.2.2 Definition of Hawkes process

Now let us proceed to the introduction of Hawkes process. Hawkes processes are self- and mutually-exciting point processes introduced by Hawkes in the works Hawkes (1971a,b).

Definition 2.4 (Hawkes process)**.** *A d-dimensional* ***Hawkes process*** *is a vector of counting processes* $\boldsymbol{N} = (N_i)_{i\in\{1,2,\dots,d\}}$ *with conditional intensity* $\boldsymbol{\lambda}_t = (\lambda_i)_{i\in\{1,2,\dots,d\}}$ *as follows:*

$$\lambda_i(t) = \mu_i + \sum_{j=1}^{d} \int_0^t \varphi_{ij}(t-s)\, dN_j(s)$$

where μ_i *is an exogenous intensity for the i-th counting process* N_i*, and* φ_{ij} *represents the influence of the arrivals of j-th events on the intensity of the i-th counting process. The vector of exogenous intensities is denoted by* $\boldsymbol{\mu} = (\mu_i)_{i\in\{1,2,\dots,d\}}$*.*

If $\varphi_{ij}(t) \equiv 0$ *for all* $1 \leqslant i, j \leqslant d$*, then* $\boldsymbol{N}(\cdot)$ *is a d-dimensional homogeneous Poisson process with rate* $\boldsymbol{\mu}$*.*

Example 1 (A univariate Hawkes process)**.** *Let us simulate a univariate Hawkes process with the following parameters:*

- *sum of exponential kernel* $\varphi(t) = 0.001e^{-0.1t} + 0.2e^{-t} + 0.1e^{-10t}$
- *baseline* $\mu = 1$*.*

See Figure 2.2 for a realization during 5 seconds.

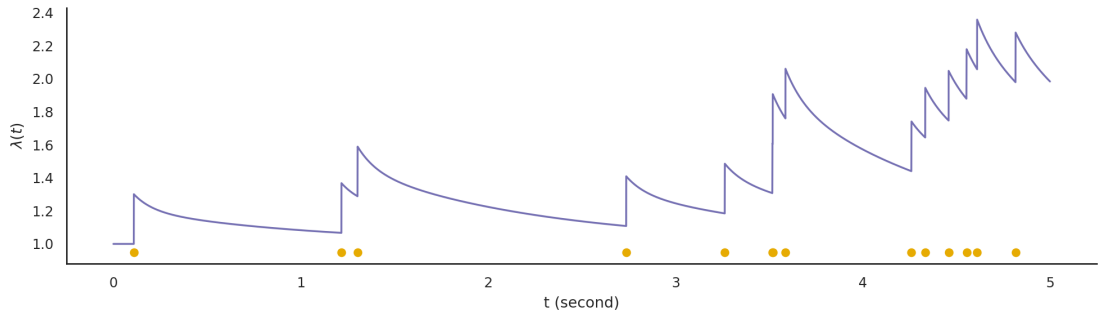


Figure 2.2 – Example of a simulated univariate Hawkes process. The orange dots represent the event times while the blue line represents the intensity function. The simulation interval is $[0, 5]$.

Remark 2.1. *If we note a multivariate Hawkes process by its event times* $\boldsymbol{T} = \{(t_k, e_k)\}_{k\in\mathbb{N}}$*, its conditional intensity can be expressed as:*

$$\lambda_i(t) = \mu_i + \sum_{k\geqslant 0} \varphi_{i,e_k}(t - t_k)\mathbb{1}_{t_k<t}$$

Example 2 (kernels)**.** *The following functions are some well-known kernels.*

- *Exponential kernel :* $\varphi_{ij}(t) = \alpha_{ij}e^{-\beta_{ij}t}(\alpha_{ij}, \beta > 0)$
- *Power-law kernel :* $\varphi_{ij}(t) = \alpha_{ij}t^{-\beta_{ij}}(\alpha_{ij}, \beta > 0)$

Proposition 2.1 (Markov property for exponential kernels)**.** *Consider a Hawkes process with exponential kernel* $\varphi_{ij}(t) = \alpha_{ij}\beta e^{-\beta t}$*. In this case,* $(\boldsymbol{\lambda}(t), \boldsymbol{N}(t))$ *is a Markov process and can be represented in a Markovian form as*

$$d\lambda_{i,t} = -\beta\lambda_{i,t}\, dt + \sum_{j=1}^{d} \alpha_{ij}\beta\, dN_{j,t}$$

For the purpose of clarity, we use the following symbols:

- $\boldsymbol{\mu}$ represents the vector of exogenous intensities, with μ_i as the intensity for event type $i = 1, 2, ..., d$.
- $\boldsymbol{\varphi}(\cdot)$ represents the matrix of kernel functions, with $\varphi_{ij}(\cdot)$ as the kernel function between event types i and j.

One of the generalizations of the Hawkes process is the non-linear case, with the intensity function expressed as :

$$\lambda_i(t) = h(\mu_i + \sum_{j=1}^{d} \int_0^t \varphi_{ij}(t-s)\, dN_j(s))$$

where $h(\cdot)$ is a non-linear positive function. The nonlinear Hawkes process has been studied by several authors, including Brémaud and Massoulié (1996) and Zhu (2013).

2.2.3 Some properties related to Hawkes processes

The linear Hawkes process has been extensively studied and documented in the literature. In the following part, we will review some key properties of the linear Hawkes process.

Proposition 2.2 (Non-explosiveness)**.** *Consider a point process* $\boldsymbol{T} = \{(t_k, e_k), k \in \mathbb{N}\}$*, set* $t_\infty = \lim_{k\to\infty} t_k$*.* $\boldsymbol{T}$ *is called non-explosive if* $t_\infty = \infty$ *almost surely.*

A Hawkes process is non-explosive if $\int_0^t \varphi_{ij}(s)ds < \infty$ *for all* $i, j \in [1:n], t \geqslant 0$

Proposition 2.3 (Stationarity)**.** $\boldsymbol{N}_t$ *has asymptotically stationary increments and* $\boldsymbol{\lambda}_t$ *is asymptotically stationary if the kernel satisfies the* ***stability condition*** *Bacry et al. (2015):*

$$\boxed{\begin{array}{l} \text{For all } i, j,\ \|\varphi_{ij}\| = \int_0^\infty \varphi_{ij}(t)\, dt < \infty \text{ and the matrix} \\ \|\boldsymbol{\Phi}(\cdot)\| = (\|\varphi_{ij}\|)_{i,j} \text{ has spectral radius smaller than 1} \end{array}} \tag{A2}$$

where the ***spectral radius*** *of a matrix is the maximum of the absolute values of its eigenvalues.*

Next, we define the matrix of functions as

$$\boldsymbol{R}(t) = \sum_{k=1}^{+\infty} \boldsymbol{\Phi}^{\star k}(t)$$

where the symbol $\star$ stands for a pointwise convolution operator for matrices i.e., given two matrices of functions $\boldsymbol{A}(\cdot)$ and $\boldsymbol{B}(\cdot)$, $\boldsymbol{D}(\cdot) = \boldsymbol{A} \star \boldsymbol{B}(\cdot)$ is a matrix of functions defined as $\boldsymbol{D}_{ij}(t) = \sum_k \int \boldsymbol{A}_{ik}(t-s)\boldsymbol{B}_{kj}(s)ds$ and $\boldsymbol{\Phi}^{\star k}(t)$ is the k-th convolution of $\boldsymbol{\Phi}(\cdot)$ with itself. In other words, $\boldsymbol{\Phi}^{\star k}(t) = \underbrace{\boldsymbol{\Phi} \star \boldsymbol{\Phi} \star ... \star \boldsymbol{\Phi}}_{k \text{ times}}$.

Corollary 2. *Under the assumption of stationarity as defined in equation (A2), it can be easily shown that* [1]

$$\boldsymbol{R} = \|\boldsymbol{R}(\cdot)\| = \boldsymbol{I} + \sum_{k=1}^{\infty} \boldsymbol{\Phi}^k = (\boldsymbol{I} - \boldsymbol{\Phi})^{-1}$$

1. By a slight abuse of notation, we use $\boldsymbol{\Phi}$ (resp. $\boldsymbol{R}$) to denote the integrated matrix $\|\boldsymbol{\Phi}(\cdot)\|$ (resp. $\|\boldsymbol{R}(\cdot)\|$) in this text.

First and second order properties

We now proceed to introduce the cumulants densities associated with a Hawkes process. The unconditional intensity (first order cumulant) $\mathbf{\Lambda} dt = \mathbb{E}[d\boldsymbol{N}_t]$ is

$$\mathbf{\Lambda} = \mathbb{E}[\boldsymbol{\lambda}_t] = \boldsymbol{R}\boldsymbol{\mu} = (\boldsymbol{I} - \mathbf{\Phi})^{-1}\boldsymbol{\mu} \tag{2.2.1}$$

And if we denote by $\boldsymbol{C}(s)\, dt\, ds = \mathbb{E}[d\boldsymbol{N}_t\, d\boldsymbol{N}_{t+s}^T] - \mathbf{\Lambda}\mathbf{\Lambda}^T\, dt\, ds$ the infinitesimal covariance matrix, then its integrated cumulant (second order cumulant) is written as

$$\boldsymbol{C} = \boldsymbol{R}\mathbf{\Sigma}\boldsymbol{R}^T \tag{2.2.2}$$

where $\mathbf{\Sigma}$ is the diagonal matrix with $\mathbf{\Lambda}$ as its diagonal values, such that $\mathbf{\Sigma}^{ii} = \mathbf{\Lambda}^i$.

We define the conditional intensity matrix

$$g_{ij}(t) = \frac{\mathbb{E}[dN_{i,t} \,|\, dN_{j,0} = 1] - \lambda_i\, dt}{dt}, \text{ for } t > 0$$

This matrix is linked to the infinitesimal covariance matrix through the following equation: $\boldsymbol{C}(t) = \mathbf{\Sigma} g(t)$. Furthermore, we have the following proposition (Bacry and Muzy, 2016):

Proposition 2.4 (Wiener-Hopf Equation)**.** *Under the assumption of stationarity as defined in equation* (A2)*, the matrix function* $\mathcal{X}(t) = \mathbf{\Phi}(t)$ *is the unique solution of the Wiener-Hopf system*

$$g(t) = \mathcal{X}(t) + \mathcal{X} \star g(t), \quad \forall t > 0$$

Higher order

In general, the cumulant of order n of a n-dimensional random vector $\boldsymbol{x} = \{x_i\}_{i\in[n]}$, denoted by $K(\boldsymbol{x})$, is defined as

$$K(\boldsymbol{x}) = \sum_{\pi}(|\pi| - 1)!(-1)^{|\pi|-1} \prod_{B\in\pi} \mathbb{E}\left[\prod_{b\in B} x_b\right]$$

where π is a partition of the set $[n]$, $|\cdot|$ denotes the cardinality of a set. For a given multi-index $\boldsymbol{i} = \{i_1, i_2, ..., i_n\} \in [d]^n$, and a given time vector $\boldsymbol{t} = \{t_1, t_2, ..., t_n\}$. The n-th order cumulant density of the Hawkes process is defined as

$$K_{\boldsymbol{i}}(\boldsymbol{t}) = \frac{K(dN_{i_1,t_1}, dN_{i_2,t_2}, \ldots, dN_{i_n,t_n})}{dt_1\, dt_2 \ldots dt_n} \tag{2.2.3}$$

where dN_{i_j,t_j} is the infinitesimal increment of the Hawkes process for the i_j-th dimension at time t_j. Jovanović et al. (2015) applied the Poisson cluster representation of Hawkes process to compute the integrated cumulant.

Limit theorems

The following limit theorems results, proved in Bacry et al. (2013b), describe the asymptotic behavior of Hawkes processes as the number of events grows to infinity.

Proposition 2.5 (Law of large numbers)**.** *Assume* (A2) *holds. Then* $\boldsymbol{N}_t \in \mathbb{L}^2(\mathbb{P})$ *for all* t *and*

$$\sup_{v\in[0,1]} \left\| \frac{\boldsymbol{N}_{Tv}}{T} - v\mathbf{\Lambda} \right\| \xrightarrow{\mathbf{T}\to\infty} \mathbf{0} \text{ almost-surely and in } \mathbb{L}^{\mathbf{2}}(\mathbb{P})$$

Proposition 2.6 (Central Limit Theorem)**.** *Assume* (A2) *holds. Then for* $v \in [0,1]$*,*

$$\frac{1}{\sqrt{T}}(\boldsymbol{N}_{Tv} - \mathbb{E}[\boldsymbol{N}_{Tv}]) \xrightarrow{T\to\infty} \boldsymbol{R}\boldsymbol{\Sigma}^{1/2}W_v \text{ in law for the Skorokhod topology}$$

where $(W_v)_{v\in[0,1]}$ *is a standard n-dimensional Brownian motion and* $\boldsymbol{\Sigma}$ *is the diagonal matrix with* $\boldsymbol{\Lambda}$ *as its diagonal values, such that* $\boldsymbol{\Sigma}_{ii} = \Lambda_i$*.*

If moreover the kernel matrix $\boldsymbol{\Phi}(\cdot) = (\varphi_{ij}(\cdot))$ *satisfies* $\int_0^\infty \sqrt{t}\varphi_{ij}(t)dt < \infty$*, then*

$$\frac{1}{\sqrt{T}}(\boldsymbol{N}_{Tv} - \boldsymbol{\Lambda}Tv) \xrightarrow{T\to\infty} \boldsymbol{R}\boldsymbol{\Sigma}^{1/2}W_v \text{ in law for the Skorokhod topology}$$

2.2.4 Simulation methods

Various methods are proposed in the literature for simulating Hawkes processes, including the ones that we will briefly introduce below. In this part, we will introduce several techniques to simulate a d-dimensional Hawkes process $\boldsymbol{N}$, on the interval $[0,T]$, associated with the intensity function

$$\lambda_i(t) = \mu_i + \sum_{j=1}^{d} \int_0^t \varphi_{ij}(t-s)\, dN_j(s)$$

Thinning

The thinning algorithm is a well-known method for simulating non-homogeneous Poisson processes and has been applied to simulate Hawkes processes as well. Originally introduced by Lewis in Lewis and Shedler (1979), the algorithm was later adapted by Ogata in Ogata and Akaike (1982) to be suitable for simulating Hawkes processes.

To simulate a non-homogeneous Poisson processes with intensity function $\lambda(t)$, the thinning algorithm consists of the following steps: (i) simulate a homogeneous Poisson process with intensity $\lambda^\star = \sup_{t\in[0,T]} \lambda(t)$, and (ii) for each point t_m generated in step (i), accept it with probability $\lambda(t_m)/\lambda^\star$, and reject it otherwise.

The thinning algorithm can be easily extended to simulate a d-dimensional Hawkes process $\boldsymbol{N}$ by adjusting the intensity function stochastically.

Cluster algorithm

The simulation of Hawkes process can be realized following a recursive branching structure (Hawkes, 1973; Rasmussen, 2013).

- For each $1 \leqslant i \leqslant d$, let $C_i^{(0)} = \{(t_m^{(0)}, i)\}_m$ be a realization on $[0,T]$ of a homogeneous Poisson process with rate μ_i. These initial points are usually called the immigrants (or generation 0), the rest of the points are called offspring.
- Then to generate the next generation, we follow an iterative process. Let $n \in \mathbb{N}^+$ be the current generation, and let $C_j^{(n)}$ be the set of events of type j in generation n.
  - For each offspring $(t_m^{(n)}, i)$ of type i, generate a sequence of first-generation events of type j, $C_j^{(n+1)} = \{(t_{m'}^{(n+1)}, j)\}_{m'}$ on the time interval $[t_m^{(n)}, T]$, with non-homogenous Poisson process with rate $\varphi_{ji}(t - t_m)$

- Repeat this process for subsequent generations until no more events are generated on $[0, T]$.

The superposition of all immigrants and offspring, $\bigcup_{n=0}^{\infty}\bigcup_{j=1}^{d} C_j^{(n)}$, constitute a realization of the corresponding Hawkes process on the interval $[0, T]$.

Time-change

If N_t is a nonhomogeneous Poisson process with intensity function $\lambda(t)$ and its cumulative intensity function $F(\cdot)$ is defined as $F(t) = \int_0^t \lambda(u)du$, $N_{F^{-1}(t)}$ is an homogeneous Poisson process of intensity 1. As a result if $(t_i)_{i\in\mathbb{N}^+}$ is a realization of Poisson process of intensity 1, then $(F^{-1}(t_i))_{i\in\mathbb{N}^+}$ is a realization of the non-homogenous process.

This method is also valid when N_t is a Hawkes process, while the greatest difficulty lies in the computation of the inverse cumulative intensity function $F^{-1}(\cdot)$. When the kernel function is restricted to exponential, there exists an analytical expression for the cumulative intensity function F, as described in Dassios and Zhao (2013).

2.2.5 Estimation

Most of the simulation and estimation techniques discussed in this thesis for Hawkes processes are accessible through the open-source Tick Python library for convenient implementation and utilization (Bacry et al., 2017). In this section, we provide a brief overview of some estimation methods for Hawkes processes. Additional estimation approaches can be found in the literature, such as least squares estimation (Reynaud-Bouret and Schbath, 2010), the INAR method (Kirchner, 2017), gradient-based methods (Cartea et al., 2021).

Given an observed sequence $(N_i)_{i=1}^d$ over a time interval of $[0, T]$, represented as a sequence of length n denoted as $(t_k, e_k), k = 1, 2, ..., n$, various methods can be employed as outlined below.

a. Maximum Likelihood Estimation (MLE)

First introduced by Ogata and Akaike (1982), the log-likelihood function is given by

$$L(\boldsymbol{\mu}, \varphi) = \sum_{i=1}^{d} L_i(\boldsymbol{\mu}, \varphi)$$

where

$$\begin{aligned} L_i(\boldsymbol{\mu}, \varphi) &= \int_0^T \log(\lambda_i(t))\, dN_i(t) - \int_0^T \lambda_i(t)\, dt \\ &= \sum_{k\in\mathbb{N}, e_k=i} \log\left(\mu_i + \sum_{j=1}^{d}\int_0^{t_k} \varphi_{ij}(t_k - s)\, dN_j(s)\right) - \mu_i T - \int_0^T \int_0^t \sum_{j=1}^{d} \varphi_{ij}(t-s)\, dN_j(s)\, dt \\ &= \sum_{k\in\mathbb{N}, e_k=i} \log\left(\mu_i + \sum_{m\in\mathbb{N}, t_m<t_k} \varphi_{i,e_m}(t_k - t_m)\right) - \mu_i T - \int_0^T \sum_{m\in\mathbb{N}, t_m<t} \varphi_{ij}(t - t_m)\, dt \end{aligned}$$

If we parameterize the Hawkes kernels φ by θ, i.e., $\varphi = \varphi_\theta$, then the MLE of $(\boldsymbol{\mu}, \theta)$ based on the observation is given by

$$\hat{\boldsymbol{\mu}}, \hat{\theta} = \arg\min_{\boldsymbol{\mu},\theta} L(\boldsymbol{\mu}, \varphi_\theta)$$

b. Expectation-Maximization (EM)

In the work by Veen and Schoenberg (2008), the authors present an algorithm based on Expectation-Maximization (EM) for parameter estimation in a Hawkes process. The EM approach introduces a latent variable u_{kl} to encode the branching structure of the Hawkes process. u_{kl} takes the value 1 if event t_k is a child of event t_l, and 0 otherwise. The likelihood conditional on u_{kl}, denoted as $L(\theta, u)$, is called the complete data likelihood. The following gives the EM algorithm process during the nth iteration:

- **E-step**: estimate the probability of u_{kl} for each pair of events (t_k, t_l) $(k > l)$, based on the current parameter $\theta^{(n)}$.
- **M-step**: maximize the expected complete data log-likelihood $L(\theta, u^{(n)})$ with respect to θ, to obtain $\theta^{(n+1)}$.

Later, a non-parametric EM estimation is studied in Lewis and Mohler (2011).

c. Wiener-Hopf

Bacry et al. (2015) proposed a non-parametric estimation method by solving numerically the Wiener-Hopf system in Proposition 2.4. The procedure is as follows:

- Estimate empirically the mean intensity $\hat{\Lambda}$ and compute the empirical conditional intensity $\hat{g}_{ij}(t)$ by using a fine grid of t values
- Solve the Wiener-Hopf system using a quadrature method. Assume that the kernels are piecewise constant on $[t^{(k)}, t^{(k+1)}]$ where $\{t^{(k)}\}_{k=1,\dots,K}$ is a partition of $[0, T]$. Then the Wiener-Hopf system can be written as a linear system of equations

$$\hat{g}_{ij}(t^{(n)}) = \varphi_{ij}(t^{(n)}) + \sum_{l=1}^{d} \sum_{k=1}^{K} (t^{(k+1)} - t^{(k)}) \hat{g}_{il}(t^{(n)} - t^{(k)}) \varphi_{lj}(t^{(k)})$$

- Solve this Kd^2 linear system and we obtain the estimated kernels $\hat{\varphi}_{ij}(t)$ as well as their norms $\|\hat{\varphi}_{ij}\|$.
- Estimate the exogenous intensity $\hat{\mu}$ through (2.2.1).

d. Non Parametric Hawkes Cumulant method (NPHC)

Achab et al. (2017) developed an estimation technique by using the cumulants method (as shown in Equation (2.2.3)). Specifically, they provided an consistent estimator for the first, second and third-order integrated cumulants and then usd these cumulants to estimate the kernel matrix norm $\boldsymbol{\Phi}$ (or equivalently $\boldsymbol{R}$ as $\boldsymbol{R} = (\boldsymbol{I} - \boldsymbol{\Phi})^{-1}$).

$$\begin{aligned}
\Lambda_i &= \lim_{\delta \to 0} \frac{1}{\delta} \mathbb{E}[N_{i,t+\delta} - N_{i,t}] = \sum_{m=1}^{d} R_{im} \mu_m \\
C_{ij} &= \lim_{\delta \to 0} \frac{1}{\delta} \int_{\tau \in \mathbb{R}} Cov(N_{i,t+\delta} - N_{i,t}, N_{j,t+\delta+\tau} - N_{j,t+\tau}) \\
&= \sum_{m=1}^{d} \Lambda_m R_{im} R_{jm} \qquad (2.2.4) \\
K_{ijk} &= \lim_{\delta \to 0} \frac{1}{\delta} \int_{\tau \in \mathbb{R}} \int_{\tau' \in \mathbb{R}} Skew(N_{i,t+\delta} - N_{i,t}, N_{j,t+\delta+\tau} - N_{j,t+\tau'}, N_{k,t+\delta+\tau'} - N_{k,t+\tau}) \\
&= \sum_{m=1}^{d} (R_{im} R_{jm} C_{km} + R_{im} C_{jm} R_{km} + C_{im} R_{jm} R_{km} - 2\Lambda_m R_{im} R_{jm} R_{km})
\end{aligned}$$

where $Skew$ stands for the function of coskewness i.e., $Skew(X,Y,Z) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])(Z - \mathbb{E}[Z])]$ for three random variables X, Y, Z.

And the estimator of these three cumulants are given by (N in the following equations stands for the observed processes):

$$\begin{aligned}
\widehat{\Lambda}_i =& \frac{1}{T}\sum_{m=1}^{n} \mathbb{1}_{e_m=i} = \frac{N_{i,t}}{T} \\
\widehat{C}_{ij} =& \frac{1}{T}\sum_{m=1}^{n} \mathbb{1}_{e_m=i}\left(N_{j,t_m+H} - N_{j,t_m-H} - 2H\widehat{\Lambda}_j\right) \\
\widehat{K}_{ijk} =& \frac{1}{T}\sum_{m=1}^{n} \mathbb{1}_{e_m=i}\left[\left(N_{j,t_m+H} - N_{j,t_m-H} - 2H\widehat{\Lambda}_j\right)\left(N_{k,t_m+H} - N_{k,t_m-H} - 2H\widehat{\Lambda}_k\right)\right] \\
& - \frac{\widehat{\Lambda}_i}{T}\sum_{p,q=1}^{n} \mathbb{1}_{e_p=j,e_q=k}(2H - |t_p - t_q|)^+ + 4H^2\widehat{\lambda}_i\widehat{\Lambda}_j\widehat{\Lambda}_k
\end{aligned} \tag{2.2.5}$$

for a H such that :

- each kernel φ_{ij} is essentially supported by $[0, H]$
- large enough s.t. the integration in the Eq 2.2.4 can pass to $[-H, H]$ with a small error
- small enough compared to T

And the estimator of $\boldsymbol{R}$ is given by $\hat{\boldsymbol{R}} \in \arg\min_{\boldsymbol{R}} \mathcal{L}(\boldsymbol{R})$ where $\mathcal{L}(\boldsymbol{R})$ is the loss function

$$\mathcal{L}(\boldsymbol{R}) = \kappa\|\boldsymbol{C}(\boldsymbol{R}) - \widehat{\boldsymbol{C}}\|_F^2 + (1-\kappa)\|\boldsymbol{K}^c(\boldsymbol{R}) - \widehat{\boldsymbol{K}^c}\|_F^2 \tag{2.2.6}$$

where

- $\|\cdot\|_F^2$ is the Frobenius norm, i.e., $\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n A_{ij}^2$ for a matrix $A = (A_{ij})$
- $\boldsymbol{K}^c = (\boldsymbol{K}_{iij})_{i,j=1}^d$, $\widehat{\boldsymbol{K}^c} = (\widehat{\boldsymbol{K}}_{iij})_{i,j=1}^d$
- $\boldsymbol{C}(R)$ and $\boldsymbol{K}(R)$ are given by Eq (2.2.4)
- $\widehat{\boldsymbol{C}}$ and $\widehat{\boldsymbol{K}^c}$ are given by Eq (2.2.5)
- $\kappa \in [0,1]$ is used to re-scale the two loss terms in Eq (2.2.6), $\kappa = \dfrac{\|\widehat{\boldsymbol{K}^c}\|_F^2}{\|\widehat{\boldsymbol{C}}\|_F^2 + \|\widehat{\boldsymbol{K}^c}\|_F^2}$

For a more comprehensive understanding of the estimation technique, we invite readers to refer to the paper Achab et al. (2017). Furthermore, one can also find the applications of this technique in finance in their subsequent publication (Achab et al., 2018).

e. Neural models

In recent years, the use of deep learning methods for modeling the Hawkes processes' conditional intensity function has become very popular. One notable example is the neural Hawkes process (NHP) proposed by Mei and Eisner (2017). In this work, they applied a continuous-time LSTM networks to model a multivariate point process. Specifically, the intensity function $\lambda_i(t)$ is obtained by

$$\lambda_i(t) = f_i(W_i^T h(t))$$

where f_i is a non-linear function and $h(t)$ is the hidden state vector of the LSTM network at time t. The hidden state vector $h(t)$ is given by $h(t) = o_k \odot (2\sigma(2c(t) - 1))$ for $t \in (t_{k-1}, t_k)$, where the intensity decay over time is encoded in the memory cell $c(t)$.

Another approach to modeling the Hawkes process is through the use of self-attention-based model, such as the one proposed in Zhang et al. (2020). In their work, each event is encoded to a representation vector $\boldsymbol{x}_k$, which is obtained by processing an embedding layer on the event type e_k and a time-shifted positional encoding. They then used a hidden vector to summarize the impact of all previous events on a given event type e'. This hidden vector $h_{e',k+1}$ is given by

$$h_{e',k+1} = \left[\sum_{j=1}^{k} f(\boldsymbol{x}_{k+1}, \boldsymbol{x}_j) g(\boldsymbol{x}_j) \right] / \sum_{j=1}^{k} f(\boldsymbol{x}_{k+1}, \boldsymbol{x}_j)$$

where $\boldsymbol{x}_{k+1}$ is like query in the attention terminology, $\boldsymbol{x}_j$ is the key and $g(\boldsymbol{x}_j)$ is the value. The function f is a similarity function, defined as $f(\boldsymbol{x}_{k+1}, \boldsymbol{x}_j) = \exp(\boldsymbol{x}_{k+1} \boldsymbol{x}_j^T)$.

Other related works in this area include the Recurrent Marked Temporal Point Process (RMTPP) proposed by Du et al. (2016), the transformer Hawkes process model (THP) proposed by Zuo et al. (2020).

2.3 - Artificial Neural Networks

In this section, we will provide a brief overview of some fundamental artificial neural networks (ANNs) that are used in this thesis. ANNs are computing systems that mimics the manner in which biological neurons communicate with each other. They are a subset of machine learning and a crucial component of deep learning.

An ANN is composed of interconnected nodes, called artificial neurons, that work together to process and transmit information. The nodes are organized into layers, including an input layer, one or more hidden layers, and an output layer. The signals are transferred from one layer to the next. By adjusting the weights of the neurons, an ANN can be trained to perform a wide range of tasks such as image classification, speech recognition etc.

In this section, we will introduce some of the most popular neural network structures.

2.3.1 Multilayer perception

A multilayer perceptron (MLP) is a fully connected class of feedforward artificial neural network (ANN).

To describe mathematically a MLP with n hidden layers, we can use the following formulation.

$$\begin{aligned} \text{Input} &: \boldsymbol{x} = \boldsymbol{h}^{(0)} \\ i\text{-th hidden layer} &: \boldsymbol{h}^{(i)} = \sigma^{(i)}(W^{(i)} \boldsymbol{h}^{(i-1)} + b^{(i)}) \\ \text{Output} &: \boldsymbol{y} = \sigma^{(n+1)}(W^{(n+1)} \boldsymbol{h}^{(n)} + b^{(n+1)}) \end{aligned}$$

where $W^{(i)}$ and b^i are the neural weights and biases. $\sigma^{(i)}(\cdot)$ is the nonlinear activation function.

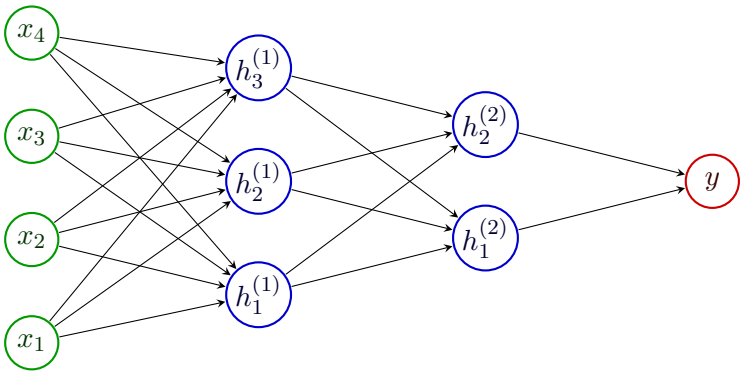


Figure 2.3 – A multilayer perceptron with 2 hidden layers and a scalar output

Activation function The following are some of the most widely used non-linear activation functions.

$$\text{Sigmoid} : \sigma(x) = \frac{1}{1+e^{-x}}$$
$$\text{Hyperbolic tangent} : tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$
$$\text{Rectified linear unit} : ReLU(x) = \max(0, x)$$

2.3.2 Recurrent Neural Network

A recurrent neural network (RNN) is a class of artificial neural networks which are widely used to time-series data and other sequential data.

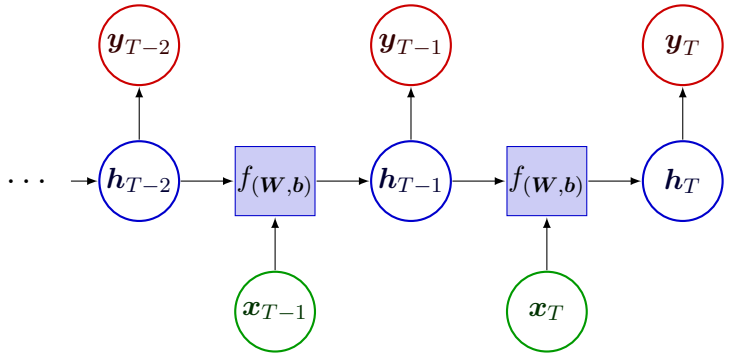


Figure 2.4 – An illustrative example of a RNN with one hidden layer. The outputs except the last one are dashed, as ...

$$\text{Input} : (\boldsymbol{x}_t)_{t=1,2,\ldots,T}$$
$$\text{Hidden state} : \boldsymbol{h}_t = f_{(W,b)}(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$$
$$= \sigma_h(W_h \boldsymbol{h}_{t-1} + W_x \boldsymbol{x}_t + b_x)$$
$$\text{Output} : \boldsymbol{y}_t = \sigma_y(W_y \boldsymbol{h}_t + b_y)$$

where W_x, W_h, W_y represent respectively the linear weights associated with inputs $\boldsymbol{x}_t$, the previous hidden state $\boldsymbol{h}_{t-1}$ and the current hidden state $\boldsymbol{h}_t$ for output. b_x and b_y are the bias terms while $\sigma_h(\cdot)$ and $\sigma_y(\cdot)$ are the nonlinear activation functions.

However in practice, RNNs often struggle with the vanishing gradient problem, which hinders their ability from keeping long-term memory. To overcome this problem,some special kinds of RNN, such as gated RNNs, have been developed. The most used in practical applications are the Long short-term memory (LSTM) and Gated Recurrent Unit (GRU).

Introduced by Hochreiter and Schmidhuber (1997), **Long short-term memory** networks (or simply LSTMs) are a special kind of RNN which were explicitly designed to avoid the vanishing gradient problem. The difference between a standard RNN and an LSTM lies in the repeating module. Instead of having a single neural network layer, LSTM has four: a cell, an input gate, an output gate and a forget gate.

$$\begin{aligned}
\text{Forget gate}: \quad & f_t = \sigma(W_f \boldsymbol{x}_t + U_f \boldsymbol{h}_{t-1} + b_f) \\
\text{Input gate}: \quad & i_t = \sigma(W_i \boldsymbol{x}_t + U_i \boldsymbol{h}_{t-1} + b_i) \\
& \tilde{C}_t = tanh(W_C \boldsymbol{x}_t + U_C \boldsymbol{h}_{t-1} + b_C) \\
\text{Cell state}: \quad & C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
\text{Output gate}: \quad & o_t = \sigma(W_o \boldsymbol{x}_t + U_o \boldsymbol{h}_{t-1} + b_o) \\
\text{Hidden state}: \quad & \boldsymbol{h}_t = o_t \odot tanh(C_t)
\end{aligned}$$

where W, U, b are weight matrices and bias vectors. $\sigma(\cdot)$ represents the sigmoid function, $tanh(\cdot)$ represents the hyperbolic tangent function and the operator $\odot$ denotes the Hadamard product (or element-wise product), i.e., for two matrices A and B, $A \odot B = (A_{ij} B_{ij})_{ij}$.

Gated Recurrent Unit (GRU) was introduced by Cho et al. (2014) in 2014, aiming to solve the vanishing gradient problem. The GRU is like an LSTM but has fewer parameters.

$$\begin{aligned}
\text{Update gate}: \quad & z_t = \sigma(W_z \boldsymbol{x}_t + U_z \boldsymbol{h}_{t-1} + b_z) \\
\text{Reset gate}: \quad & r_t = \sigma(W_r \boldsymbol{x}_t + U_r \boldsymbol{h}_{t-1} + b_r) \\
& \tilde{h}_t = tanh(W_h \boldsymbol{x}_t + U_h(r_t \odot \boldsymbol{h}_{t-1}) + b_h) \\
\text{Hidden state}: \quad & \boldsymbol{h}_t = z_t \odot \boldsymbol{h}_{t-1} + (1 - z_t) \odot \tilde{h}_t
\end{aligned}$$

where the variable and function notations are similar to those in LSTM.

Bidirectional RNN Bidirectional RNNs (BiRNNs) are a type of neural networks that extend the standard RNN by including an additional RNN layer. Unlike unidirectional RNNs, BiRNN allows that the input flows in two directions such that the output layer has access to information from both past and future. Figure 2.5 provides a visual representation of the unrolled Bidirectional Recurrent Neural Network. [2]

2.3.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a class of artificial neural networks that are most commonly applied to analyze visual imagery. In contrast to traditional neural networks that use general matrix multiplications, CNNs apply convolution operations in at least one of their layers.

2. Let note that there is a slight trick in this figure, the outputs of the backward layer will be reversed in fact.

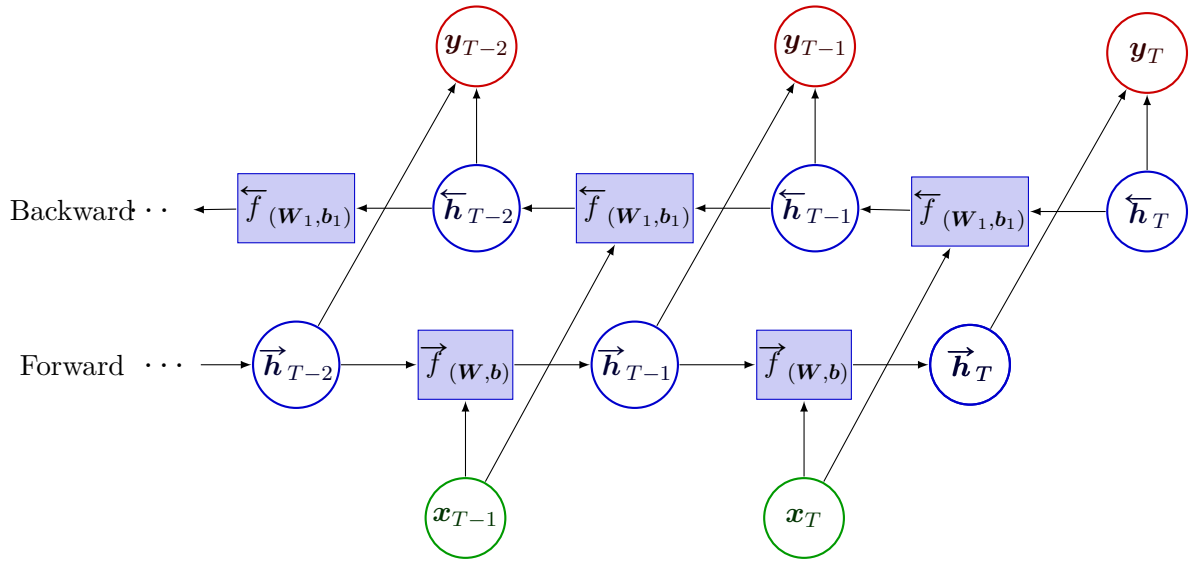


Figure 2.5 – An illustrative example of a Bidirectional RNN.

A basic CNN consists of a sequence of layers, similar to the MLP, with three main types of layers: convolutional layer, pooling layer and fully connected layer.

The **convolutional layer** is a crucial block of the CNN architecture. It contains a set of filters or kernels (as shown in Figure 2.6), which connect local regions in the input. By applying the filters to the input data, the convolutional layer can detect or enhance some features.

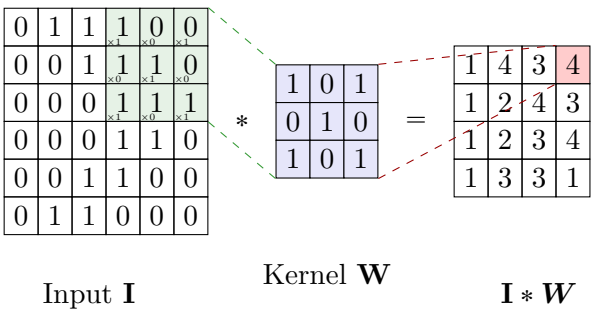


Figure 2.6 – An illustrative example of a convolutional layer

The **Pooling layers** are responsible for reducing the dimensions of data by combining the groups of the outputs into a single neuron. Max pooling and Average pooling are the most common pooling operations. As indicated in their names, Max pooling returns the maximum value of a local group of neurons which Average pooling takes the average value.

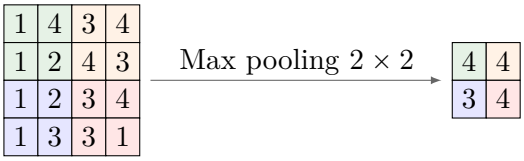


Figure 2.7 – An illustrative example of a pooling layer

The **Fully connected layer** is the same as a traditional MLP, it connects every neuron in the current layer to every neuron in the next layer.

CHAPTER 3

STATE-DEPENDENT SPREAD HAWKES MODEL

From: The self-exciting nature of the bid-ask spread dynamics (Ruan et al., 2023b)
R. Ruan, E. Bacry, J.-F. Muzy

The bid-ask spread, which is defined by the difference between the best selling price and the best buying price in a Limit Order Book at a given time, is a crucial factor in the analysis of financial securities. In this study, we introduce the "State-dependent Spread Hawkes model" (SDSH), a new Hawkes process model for spread dynamics that accounts for various spread jump sizes and incorporates the impact of the current spread state on its intensity functions. Through the application of the SDSH model to high-frequency data from the CAC40 Euronext market, we demonstrate its efficacy in capturing diverse statistical properties, including the spread distributions, inter-event time distributions, and spread autocorrelation functions. Furthermore, we illustrate the ability of the SDSH model to forecast spread values at short-term horizons.

Contents

3.1 - Introduction

The bid-ask spread, defined at a given time in a Limit Order Book by the difference between the smallest ask (selling) price and the largest bid (buying) price, is a quantity of great interest for financial securities. It represents the cost of an "immediate" transaction rather than a more patient one. Many studies in economics literature are devoted to the bid-ask spread which is in general decomposed into order processing costs, adverse selection costs and inventory risk (for the liquidity providers/market makers who earn money from the spread) (Glosten and Harris, 1988; Glosten and Milgrom, 1985; Huang and Stoll, 1997; Stoll, 1989). The spread is often used as a proxy for market liquidity, with narrower spreads commonly associated to highly liquid markets. For a literature review of liquidity measures, we refer readers to Díaz and Escribano (2020); Fong et al. (2017); Fosset et al. (2020b); Goyenko et al. (2009). Let us also mention that the bid-ask spread has been proved to be very closely related to realized volatility. Indeed, many empirical studies (Fong et al., 2017; Goyenko et al., 2009) consistently prove a robust positive correlation between these two variables (Bessembinder, 1994; Dayri and Rosenbaum, 2015; Wyart et al., 2008; Zumbach, 2004).

Many other statistical properties of the bid-ask spread have been the focus of various works. For instance, it has been shown that spread has a fat-tailed distribution and its dynamics is characterized by a long range (power-law) auto-correlation function (Bouchaud et al., 2018, 2009; Fall et al., 2021; Groß-KlußMann and Hautsch, 2013; Mike and Farmer, 2008; Plerou et al., 2005). The studies referenced (Mike and Farmer, 2008; Ponzi et al., 2006; Zawadowski et al., 2006) show that after a large variation of the spread (i.e., a temporary liquidity crisis), the spread decays slowly back to an equilibrium value. Let us note that, even though most of the time these statistical properties are obtained through direct empirical studies on historical spread time-series, some papers tackle the statistical properties of the spread (mainly the distribution of the spread values) via some statistical models of the limit and market order flows (Abergel and Jedidi, 2013; Bouchaud et al., 2002; Daniels et al., 2003; Foucault et al., 2005; Muni Toke and Yoshida, 2017; Roşu, 2009; Smith et al., 2003). Readers can find a nice overview of the different statistical properties of the bid-ask spread in the chapter 7 of Bouchaud et al. (2009).

In order to understand the properties of market prices and their formation in the context of electronic markets, many approaches involving point processes have been proposed to describe the occurrence of order book events (see e.g., Abergel and Jedidi (2013); Cont et al. (2010); Smith et al. (2003)). Within this context, as reviewed in Bacry et al. (2015), Hawkes processes emerge as a widely adopted class of models, for their effectiveness in describing the dynamical properties of different quantities like the market activity (Bowser, 2007; Hardiman et al., 2013; Morariu-Patrichi and Pakkanen, 2022; Muni Toke and Pomponio, 2012), the mid-price (Bacry et al., 2013a), the best bid / best ask prices (Lee and Seo, 2022) and the first (L1) book levels (Bacry et al., 2016; Large, 2007; Zheng et al., 2014). Hawkes processes constitute a class of multivariate point processes that were introduced in the seventies by A.G. Hawkes (Hawkes, 1971a) notably to model the occurrence of seismic events. They involve an intensity vector that is (in its original form) a simple linear function of past events. This popularity of Hawkes processes can be explained above all by their great simplicity and flexibility.

Specifically, we would like to highlight a category of Hawkes processes referred to as "state-dependent" Hawkes processes. These processes dynamically adjust their intensity based on the current state of the system. Noteworthy contributions to state-dependent models in finance can

be found in the references Morariu-Patrichi and Pakkanen (2022); Sfendourakis and Toke (2021); Toke and Yoshida (2016); Wu et al. (2019).

Our main purpose of this chapter is to design a dynamical model for spread fluctuations based on Hawkes processes. Though many studies can be found in the literature about the bid-ask spread and its statistical properties, only a limited number provide models for its dynamics at high frequencies. Besides, some recent econometric approaches involving long-memory auto-regressive, Poisson point processes (Cattivelli and Pirino, 2019; Groß-KlußMann and Hautsch, 2013), one can mention few models that rely on Hawkes processes. Zheng et al. (2014) proposed one of the first models for the bid-ask spread dynamics of a financial asset which uses a constrained 2-dimensional Hawkes process. The spread S_t is a positive integer multiple of the tick value (i.e., the minimum increment defined by the market between two quotation values). The first (resp. second) dimension of this Hawkes process S_t^+ (resp. S_t^-) is used for encoding the positive (resp. negative) jumps of the spread. S_t^+ (resp. S_t^-) is an increasing function on time, with its value jumping by 1 every time a positive(resp. negative) jump in the spread is observed. In this model, all jumps of the spread are assumed to be 1 tick in size. Thus, one gets $S_t = S_t^+ - S_t^-$. The constrained Hawkes process of Zheng et al. (2014) is defined by the intensities λ^+ and λ^- of the respective jump processes S_t^+ and S_t^-:

$$\begin{aligned} \lambda_t^+ &= \mu^+ + \sum_{e\in\{+,-\}} \int_0^t \varphi^{+,e}(t-s) dS_s^e \\ \lambda_t^- &= 1_{S_{t-}\geqslant 2}(\mu^- + \sum_{e\in\{+,-\}} \int_0^t \varphi^{-,e}(t-s) dS_s^e) \end{aligned} \tag{3.1.1}$$

where μ^+ and μ^- correspond respectively to constant exogenous intensities (i.e., terms that do not depend on the past events) for S^+ and S^-. The 4 functions $\{\varphi^{e,e'}(t)\}_{e,e'=\pm}$ are causal kernels (i.e., functions with support $[0,+\infty)$ that encode the endogenous influence of past jumps of type e' on the occurrence of future jumps of type e). For the purpose of estimation, the kernels are often taken to be exponential functions or a sum of exponential functions. This choice is motivated by the fact that exponential kernel functions yield an explicit likelihood function, thus enabling efficient computation. Zheng et al. (2014) chose exponential kernels of the form: $\varphi^{e,e'}(t) = \alpha^{e,e'}\beta e^{-\beta t}$ (β is the same for all the kernels). Let us point out that, Zheng et al. introduced a non-linearity (the term $1_{S_{t-}\geqslant 2}$) in the "classical" definition of a Hawkes process. This non-linearity is necessary to constrain the spread to keep from taking negative or zero values. In their work, Zheng et al. performed numerical estimation of their model (using Maximal Likelihood Estimations) and studied some statistical properties.

More recently, Fosset et al. (2020a) built a simplified version of the previous model (3.1.1) and focused on the relation between spread dynamics and liquidity crises. Their constrained 2-dimensional Hawkes process has intensity functions:

$$\begin{aligned} \lambda_t^+ &= \mu^+ + \int_0^t \alpha\beta e^{-\beta(t-s)}\, dS_s^+, \\ \lambda_t^- &= \mu^- 1_{\{S_t\geqslant 2\}}, \end{aligned} \tag{3.1.2}$$

where the notation is as in model (3.1.1). The threshold $\alpha_c = 1 - \frac{\mu^+}{\mu^-}$ distinguishes different stable regimes. When $\alpha < \alpha_c$, the Hawkes system is stable and the invariant distribution of spread is given. When $\alpha_c < \alpha < 1$, the model is still stable but the spread grows linearly. When $\alpha > 1$, the system is explosive and there is a liquidity crisis.

In this chapter, we propose an approach inspired by these two papers but our model focuses more on fitting empirical features and matching observed properties from market data. More specifically, we propose the "State Dependent Spread Hawkes" (SDSH) model which accounts for bid-ask spread fluctuations. The SDSH model can be seen as a generalization of the two previous models in that it is a $2K$-variate ($K \geqslant 1$) Hawkes process which accounts for different jump sizes (up to size K). It comprises multiple kernels to encode the influence of past jumps on future jumps. Each component includes a non-linear element, as in the previous models, in order to ensure the spread value remains strictly positive. In order to allow more complex dynamics, each kernel is a sum of exponential functions, in contrast to the previous models. Our ambition is to account for more aspects of the bid-ask spread dynamics that were not captured by the previous models.

Outline The chapter is organized as follows. The next section (Section 3.2) is devoted to a detailed definition of our new model for the bid-ask spread. The Markov and ergodicity properties of this model is studied in a particular case. Section 3.3 illustrates this model with some numerical simulations and presents the estimation procedure on these simulations. Section 3.4, the core of our chapter, is devoted to empirical results. This sections is divided in 4 subsections. The first two correspond to a quick presentation and basic statistical properties of the real financial time-series that will be used all along the section. The third one concerns the estimation of our model on these data. Parameters estimation (including the kernels and the constraint parameters) are discussed thoroughly. The last one is devoted to the goodness of fit of the model through different quantities, mainly: the inter-event time distribution, the spread distribution and the autocorrelation function. Section 3.5 provides a first approach that demonstrates the interest of the SDSH model in the issue of short-term forecasting of spread values. We conclude in section 3.6.

3.2 - State-Dependent Spread Hawkes processes

3.2.1 Notation

Consider the limit order book (LOB) associated with a given asset. We note S_t the bid-ask spread of this order book at time t, or in other words, the difference of the best ask price and the best bid price at time t. We choose to express S_t in tick units where the tick corresponds to the smallest price increment authorized by the market. S_t is therefore a right continuous process that can take only strictly positive integer values (i.e., $S_t \in \mathbb{N}^*$). Its smallest possible value is $S_t = 1$ (tick).

The process S_t can be interpreted as a jump process. Various orders sent to the LOB, depending on their types and volumes, may induce different jump sizes in S_t. In our model, an event corresponds to a specific jump in S_t, characterized by its size and direction. In practice, very large jumps are extremely rare. Therefore, without loss of generality, we can limit the set of events to $\mathcal{E} = \{+1, +2, \dots, +K, -1, -2, \dots, -K\}$, where K is a hyper-parameter of the model to be fixed for each asset.

For each event $e \in \mathcal{E}$ (i.e., for each jump type e), we denote S_t^e the counting process that counts the number of jumps of size e that occurs over time, beginning arbitrarily at time 0. Thus, we can write the spread process S_t as:

$$S_t = S_0 + \sum_{k=1,2,\dots,K} k S_t^{+k} - \sum_{k=1,2,\dots,K} k S_t^{-k}. \tag{3.2.1}$$

In real limit order books, it is clear that the dynamics of the spread jumps highly depend on the

size of the jumps. It is thus reasonable to build a model which incorporates different dynamics for different jump sizes.

3.2.2 The SDSH spread model

In our model for spread dynamics, we assume that the multivariate counting process $\{S_t^e\}_{e\in\mathcal{E}}$ follows a $2K$-variate Hawkes process. However, since the dynamics of the various events depend on the state of the current spread S_t itself (e.g., when the spread is 1 tick, no negative event can occur), we introduce an additional term within the classical Hawkes framework. Depends on S_t, this term represents a *state variable* that accounts for the current size of the spread. More precisely, if we note λ_t^e the conditional intensity (at time t) of the counting process S_t^e, the State Dependent Spread Hawkes spread model can be formulated as follows:

$$\lambda_t^e = f^e(S_{t-})\left[\mu^e + \sum_{e'\in\mathcal{E}}\int_0^t \varphi^{e,e'}(t-s)dS_s^{e'}\right] \tag{3.2.2}$$

where (following the classical Hawkes processes framework)

- μ^e is the exogenous base intensity for the jumps of size e
- $\Phi(t) = \{\varphi^{e,e'}(t)\}_{e,e'\in\mathcal{E}}$ is the Hawkes kernel matrix. The elements of this matrix are Hawkes interaction kernels which are positive valued and encode the influence of past jumps of type e' on future jumps of type e.
- we assume that these kernels can be parameterized using a sum L exponentials; that is

$$\varphi^{e,e'}(t) = \sum_{l=1}^{L} \alpha_l^{e,e'}\beta_l e^{-\beta_l t}. \tag{3.2.3}$$

  Indeed, this setting is hardly restrictive since many behavior can be easily reproduced by a sum of exponentials (Bochud and Challet, 2007).

The current value of the spread is taken into account through the global multiplicative term $f^e(S_{t-})$ where

- S_{t^-} is the the left limit of S at time t (i.e., the spread size "just before" time t),
- f^e is a non-negative function defined on $\mathbb{N}^*$. It is subject to the the constraint that $f^{-k}(n) = 0$ when $n - k \leqslant 0$, in order to prevent jumps leading to a zero or negative spread value. Let us note that the Equation (3.2.2) remains unchanged if we multiply f^e by a factor and simultaneously divide μ^e and $\varphi^{e,e'}$ by the same factor. Therefore in the following, we set arbitrarily the first non-zero value of $f^e(s)$ to 1 (i.e., $f^e(\min_s\{s,\ f^e(s) \neq 0\}) = 1$).

Let us point out that, in order to account for the impact of the current spread value on the different jump dynamics of the spread, multiple choices are available. One intuitive option would have been to make the kernel themselves depend on the spread size S_{t_-}. However, such a choice would significantly increase the number of parameters for the kernels themselves, leading to important estimation instabilities. The choice we made allows several things at once

- It keeps the number of estimation parameters at a manageable level.
- It encodes in an easy way the fact that negative values for S_t are forbidden.
- It enables the model to address the well known fact that the spread dynamics is mainly mean reverting and that the higher the spread size is, the higher is the probability for large negative jumps to occur.

Compared to the previous models (3.1.1) and (3.1.2) mentioned in Section **??**, the SDSH model introduces many features that allow it to better account for the real dynamics of the spread:

- **Flexibility in jump sizes:** Unlike the previous models, the jump sizes in our model are no longer constrained to be 1 tick.
- **State-Dependent Adjustments:** The model adjusts the intensity of each jump size not only based on previous jumps but also on the current spread value, through the introduction of "state-dependent" functions f.
- **Complex Dynamics:** The choice of the sum of exponentials for Hawkes kernels allows more complex dynamics compared to simple exponential kernels as in Fosset et al. (2020a); Zheng et al. (2014). Specifically, many existing studies highlighted that the spread dynamics are characterized by long-memory properties.

3.2.3 Markov property and ergodicity

Using Eq.(3.2.1), since the Hawkes kernels are sums of exponential functions, it is easy to prove the following property:

Proposition 3.1. *The process* (S_t, X_t)*, where* $X_t^{e,e'} := \int_0^t \varphi^{e,e'}(t-s)dS_s^{e'}$*, is a Markov process.*

In the simple scenario where $K = 1$ (i.e., only jumps of size $+1$ or -1 are allowed) and $L = 1$ (i.e., only one exponential function), one can prove the following ergodic property:

Proposition 3.2. *Assume* $K = 1$ *and* $L = 1$*, which implies that* $\mathcal{E} = \{-1, 1\}$ *and* $\varphi_{e,e'}(t) = \alpha^{e,e'}\beta e^{-\beta t}$*. The process* (S_t, X_t) *is a* ***V-uniformly ergodic Markov*** *process, if the following conditions hold true:*

$$\begin{cases} f^-(1) = 0 & (A_1) \\ f^-(n) \geqslant \gamma n \text{ for some } \gamma > 0 \text{ when } n \in \mathbb{N} \text{ and } n \geqslant 2 & (A_2) \\ \sup\limits_{n \geqslant 1}\{f^+(n)\}(\alpha^{+,-} + \alpha^{+,+}) < 1 & (A_3) \end{cases}$$

Proof. The detailed proof for this result can be found in Appendix 3.A. □

Thus, under these assumptions, the spread process possesses stationary distributions. Let us notice that Condition (A_2) ensures the spread to "return to the mean value", similar to the proportional cancellation rate condition in Abergel and Jedidi (2013); Smith et al. (2003); Wu et al. (2019). Condition (A_3) corresponds to a stability condition ensuring the number of upward spread jumps does exponentially diverge within a finite interval. This is a first result, the general proof (for any K and L) seems to be much more challenging (as indicated in Remark 3.2), it will be the focus of a forthcoming study.

An example of a function f for Proposition 3.2 is as follows:

$$f^-(n) = a(n-1) \text{ for } n \in \mathbb{N}^*,$$
$$f^+(n) \equiv \frac{1}{1 + \alpha^{+,-} + \alpha^{+,+}} \text{ for } n \in \mathbb{N}^*.$$

In theory, in order to obtain ergodicity, f^+ needs to be bounded by a positive constant value, while f^- should exhibit at least a linear growth rate approximately, with respect to the current spread

value, denoted as n. However, it is important to note that in practice, when n is large enough (let us say when n exceeds a threshold S^* for some $S^* \in \mathbb{N}^*$), the probability of the spread reaching n (i.e., $S = n$) becomes extremely low. Therefore, in practical situations, for $n \geqslant S^*$, we approximate $f^e(n)$ with the value of $f^e(S^*)$.

3.3 - Numerical simulation and parametric estimation

3.3.1 Simulation

In order to perform numerical simulations, we use the classical "thinning method" introduced by Lewis and Shedler (1979); Ogata (1981) and the Tick open source library (Bacry et al., 2017).

To make it easier for readers to understand how our model works in practice, Figure 3.1 shows the result of a simulation during 20 seconds. The parameters for this simulation are as follows:

- $K = 1$, i.e., $\mathcal{E} = \{+1, -1\} =: \{+, -\}$,
- $L = 1$ and $\varphi^{e,e'}(t) = \alpha^{e,e'} e^{-\beta t}$, where $\beta = 1$, $\alpha^{+,+} = \alpha^{-,-} = 0.1$ and $\alpha^{+,-} = \alpha^{-,+} = 0.2$,
- $\mu^+ = \mu^- = 0.3$,
- $f^+(1) = 1, f^+(2) = 0.7$ and $f^+(S) = 0.3$ for $S \geqslant 3$,
- $f^-(1) = 0, f^-(2) = 1$, and $f^-(S) = 5$ for $S \geqslant 3$.

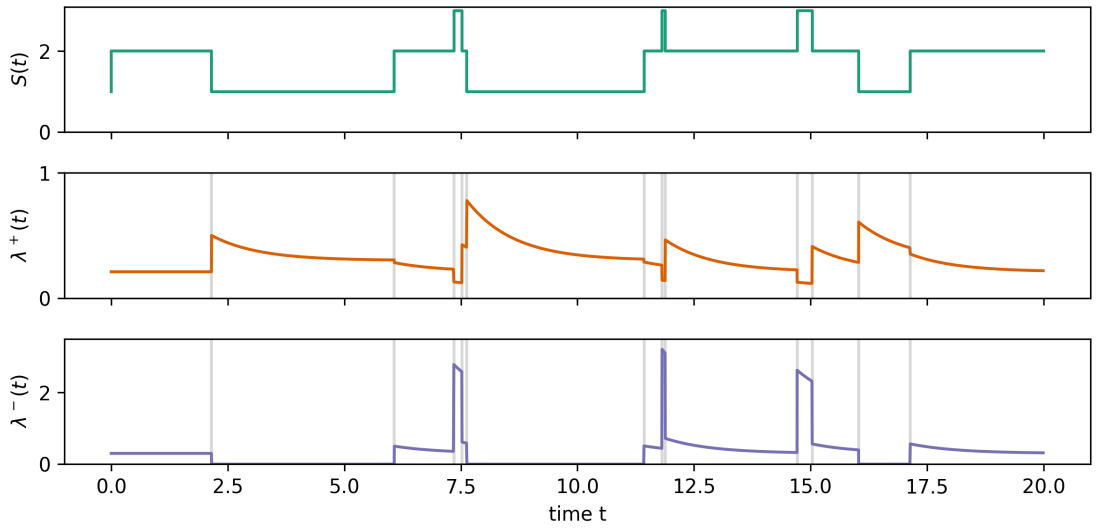


Figure 3.1 – A realization of the SDSH spread model during 20 seconds using parameters as indicated in the text. As we see, since we choose $f^-(S)$ to be very large (5) as soon as $S \geqslant 3$, the spread does not stay long at a value of 3.

3.3.2 Estimation principles

For parametric estimation, we will once again use the Tick python open source library. This library supports robust parametric Maximum Likelihood Estimation (MLE) estimation for multi-dimensional Hawkes processes using sum of exponentials. We slightly adjust the algorithms of this library in order to incorporate the $f^e(S)$ terms. The algorithm used in this model closely follows the approach used for the QRH-II model described in Wu et al. (2019). The likelihood function for this model is located in Appendix 3.B.

More precisely, to perform the parametric estimation, a set of hyper-parameters needs to be fixed beforehand, according to the specificity of each asset (as discussed in Section 3.4 for specific cases):

- K: the highest jump size allowed by the model. This hyper-parameter should be carefully chosen by looking at the effective probability of various jump sizes occurring in the real data.
- L and $\{\beta_l\}_{l=1,\dots,L}$: in practice it is sufficient to recover a very large variety of behavior by choosing the β_l such that they are logarithmically spaced. A common choice is $\beta_l = \beta_1 10^{l-1}$. L and β_1 should be chosen to match the timescale range of interest.
- $f^e(s)$: each $f^e(s)$ is a function of $s \in \mathbb{N}^*$ which corresponds to an infinite number of parameters. To make parametric estimation feasible, we have to make some assumptions on the behavior of the $f^e(s)$ functions for large s. Given that very large spreads are extremely rare events, nearly any assumption works as long as it enforces mean reversion of the spread. For our model, we arbitrarily chose to consider that all the functions $f^e(s)$ are constant for s greater that a predefined value $\bar{S}$. In practice, the value of $\bar{S}$ should be chosen so that spreads of size greater than $\bar{S}$ are extremely rare.

Given these hyper-parameters, our MLE parametric estimation algorithm allows to estimate

- the exogenous intensities $\{\mu^e\}_{e\in\mathcal{E}}$ (a total of $2K$ parameters)
- the kernel parameters $\{\alpha_l^{e,e'}\}_{l\in L,(e,e')\in\mathcal{E}^2}$ (a total of $4LK^2$ parameters)
- the values $\{f^e(s)\}_{e\in\mathcal{E},s\in[1..\bar{S}]}$ (a total of $(2K\bar{S} - \frac{K^2+K}{2} - 2K)$ parameters), under the condition that $K < \bar{S}$ (let us remind that we fixed arbitrarily the first non zero value of $f^e(s)$ to be 1).

So that amounts to $2K + 4LK^2 + 2K\bar{S} - \frac{K^2+K}{2} - 2K$ parameters.

In the following section, we illustrate the estimation procedure on simulated data.

3.3.3 Estimation on simulated data

In this example, we simulate our model in dimension 2 (i.e., $\mathcal{E} = \{+1, -1\} := \{+, -\}$), implying that the only possible movements for the spread are upward or downward shifts of one tick. The simulation process, executed through the thinning method (Lewis and Shedler, 1979; Ogata, 1981) as explained in Section 3.3.1, generates 50 independent samples of size 5000 seconds. We set the parameters for the simulation to be (using the notations introduced in Section 3.2.2):

- $\mu^+ = 0.3, \mu^- = 0.2$.
- $L = 2$ with $\beta_1 = 20s^{-1}, \beta_2 = 200s^{-1}$ and the Hawkes kernel is

$$\varphi(t) = \begin{pmatrix} 2 & 6 \\ 10 & 0 \end{pmatrix} e^{-20t} + \begin{pmatrix} 4 & 20 \\ 20 & 4 \end{pmatrix} e^{-200t}$$

  .
- $f^+(1) = 1, f^+(2) = 0.8, f^+(3) = 0.5, f^+(s) = 0.2$ for $s \geqslant 4$,
  $f^-(1) = 0, f^-(2) = 1, f^-(3) = 2, f^-(s) = 3$ for $s \geqslant 4$.

Estimation Then we estimate the parameters by employing a sum of exponential functions with $\beta_1 = 10s^{-1}, \beta_2 = 100s^{-1}, \beta_3 = 1000s^{-1}$,

$$\varphi(t) = 10 \begin{pmatrix} \alpha_1^{++} & \alpha_1^{-+} \\ \alpha_1^{+-} & \alpha_1^{--} \end{pmatrix} e^{-10t} + 100 \begin{pmatrix} \alpha_2^{++} & \alpha_2^{-+} \\ \alpha_2^{+-} & \alpha_2^{--} \end{pmatrix} e^{-100t} + 1000 \begin{pmatrix} \alpha_3^{++} & \alpha_3^{-+} \\ \alpha_3^{+-} & \alpha_3^{--} \end{pmatrix} e^{-1000t}$$

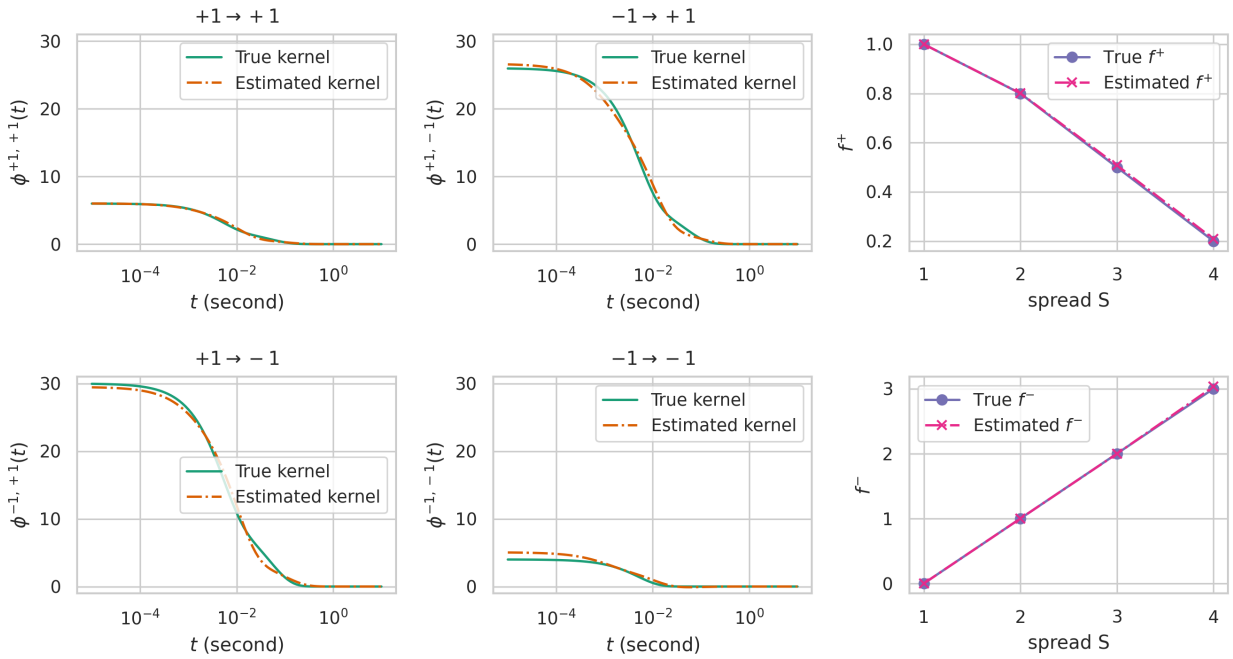


Figure 3.2 – Numerical estimation from samples of a SDSH spread model. The four figures on the left show the true kernels and the estimated kernels, and the two figures on the right show the true f values and the estimated f values. The estimated μ^+ is 0.29 and the estimated μ^- is 0.19.

As evidenced by Figure 3.2, our estimation method successfully retrieves the parameter values μ^e and accurately reproduce the shapes of both $f^e(S)$ functions and the Hawkes kernel. It is worth noting that the basis exponential functions for the simulation are very different from those for estimation.

3.4 - Empirical results

3.4.1 Data

In this section, we calibrate the SDSH spread model using high-frequency data from the CAC40 French Euronext Market. The dataset includes every single change of the spread for 3 stocks (AXA, BNP and NOKIA) and for the CAC40 index Future. The events are characterized by their jump types and the timestamps with a precision of $1\mu s$. The data for stocks (resp. CAC40 Future) are extracted from February 1st 2017 (resp. January 4th 2016) to February 28th 2018 (resp. February 28th 2017). In order to minimize the intraday seasonality of the data we used only the data in the intraday slot [10am,12am]. Thus, for a given asset, the estimation of the model is performed considering each day as an independent realization. In order to avoid days with insufficiently short time series, for a given asset, we omit all days associated with a number of events (i.e., spread changes) below a threshold. We refer readers to the Table 3.1 for more detailed information about each of these data series. It is important to note that the CAC40 Future has a much larger tick size compared to the other three assets, which leads us to expect that its spread jump sizes will be much smaller on average.

Asset	tick size	Min. #events	#days	Total #events	$\mathbb{E}_{event}[S]$	$\mathbb{E}_{cal}[S]$
CAC40 Future	0.5	5,000	100	1,026,156	1.51	1.40
AXA	0.005	3,000	130	617,309	2.44	3.04
BNP	0.01	5,500	100	1,079,464	2.54	3.17
NOKIA	0.001	2,000	108	570,754	3.52	4.43

Table 3.1 – Characteristics of the data used for model estimation. For each asset, for each day we only consider the slot from 10am to 12pm. Moreover, we keep only days where the number of events (i.e., the number of times the spread changes) is above the 'Min. #events' number. $\mathbb{E}_{event}[S]$ and $\mathbb{E}_{cal}[S]$ are the expectations of spread with two different distributions: event time distribution and calendar time distribution. See Eq. (3.4.2) and (3.4.1) for their definitions.

3.4.2 Spread distribution and hyper-parameter settings

To determine the optimal values for hyper-parameters, as discussed in Section 3.3.2, we have to examine the empirical distribution of the spread. Inspired by Bouchaud et al. (2009), we consider two methods to measure the spread distribution. Each method is constructed as an average of daily distributions.

- The first method uses the calendar time daily distribution:

$$\mathbb{P}_{cal}(S = s) = \frac{1}{T}\int_0^T 1_{S_u = s}\, du \tag{3.4.1}$$

  where $[0, T]$ corresponds to the slot 10am-12pm.

- The second method uses the event time daily distribution:

$$\mathbb{P}_{event}(S = s) = \frac{1}{N}\sum_{n=1}^{N} 1_{[S_{t_n -} = s]} \tag{3.4.2}$$

  where N is the total number of events on a given day from 10am to 12pm.

The solid lines in Figures 3.6 and 3.C.3 illustrates the so-obtained results. Let us point out that it also displays (the right column) the distribution of the spread increments (i.e., dS).

Choice of K. As expected, since the tick of the CAC40 Future is much larger than the other assets, we expect its spread jumps to be mainly of 1 tick. This expectation is indeed confirmed in the right figure of 3.C.3c, where the amplitude of spread variations dS almost never exceeds 1 tick. Consequently, it seems natural to choose $K = 1$ for this asset. If we follow the same guidelines (i.e., selecting K as the minimum value for which events corresponding to a spread change of $> K$ ticks are very rare) it seems reasonable to choose $K = 2$ for all other assets (see Figure 3.6, 3.C.3a, 3.C.3b).

Choice of $\bar{S}$. As explained in Section 3.3.2, it appears reasonable to choose arbitrarily $\bar{S}$ as the maximum value for which the probability of the spread to be this value is not extremely close to

zero (i.e., $\bar{S} = \max\{s \,|\, \mathbb{P}(S = s) \geqslant \delta\}$ for a given $\delta > 0$). In fact, it is almost impossible to perform reliable estimations of $f^e(S)$ for values of S which hardly never occur. More precisely, we choose $\bar{S}$ to be the maximum s for which $\mathbb{P}_{event}(S = s) \geqslant 1\%$. This leads to the following values: $\bar{S} = 5$ (AXA), $\bar{S} = 5$ (BNP), $\bar{S} = 8$ (Nokia) and $\bar{S} = 2$ (CAC40 Future).

Choice of L. As demonstrated in Figure b in Section 3.4.3, choosing $L = 6$ and $\beta_1 = 10^{-1}s^{-1}$ for all assets (consequently, following Section 3.3.2 $\beta_2 = 1s^{-1}$, $\beta_3 = 10s^{-1}$, $\beta_4 = 10^2 s^{-1}$, $\beta_5 = 10^3 s^{-1}$, $\beta_6 = 10^4 s^{-1}$) is sufficient to capture the kernels dynamics on the time-scale $[10^{-4}s, 10s]$. Let us point out that we perform estimations with larger β_L and smaller β_1, but these adjustments do not change significantly the results (though they increase significantly the estimation time or alternatively lead to unstable results).

Table 3.2 summarizes all the choices for the hyper parameters

Asset	K	$\bar{S}$	Kernel time scale	# parameters
CAC40 Future	1	2	$10^{-4}s \to 10^{1}s$ $(L = 6)$	27
AXA	2	5	$10^{-4}s \to 10^{1}s$ $(L = 6)$	113
BNP	2	5	$10^{-4}s \to 10^{1}s$ $(L = 6)$	113
NOKIA	2	8	$10^{-4}s \to 10^{1}s$ $(L = 6)$	125

Table 3.2 – Values chosen for the hyper parameters (following the guidelines in Section 3.3.2) for the dataset of each asset. K corresponds to the maximum jump size for the spread. $\bar{S}$ is the spread value above which the functions $f^e(s)$ are considered as constant and the kernel time scales is deduced from the choice of L and β_1

3.4.3 Estimation

We perform parameter estimation for each time-series outlined in the previous Section using MLE following the guidelines provided in Section 3.3.2. As mentioned before, all estimations in this study are performed using the Tick open-source package (Bacry et al., 2017), after having adapted the MLE exponential-kernel estimation algorithm to account the state dependent function $f^e(S)$. In the following we will present the outcomes of these estimations and provide comments on them. We start with the estimations of the $\{f^e(s)\}_{e\in\mathcal{E}}$ functions, followed by the estimations of the kernels $\{\varphi^{e,e'}(t)\}_{e,e'\in\mathcal{E}}$.

a. Estimation of the $\{f^e(s)\}_{e\in\mathcal{E}}$ functions

The results of the estimated f^e functions for different assets are displayed in Figure 3.3. Let us recall that, for each e, the first positive value of $f^e(S)$ is arbitrarily designated to 1.

An first observation is that all estimated curves exhibit the same behavior across all assets. As expected, we notice that the the f^e functions corresponding to positive events (i.e., upward jumps, $e \in \{+1, +2\}$) are decreasing functions, which decrease very quickly towards 0 whereas the f^e functions corresponding to negative events (i.e., downward jumps, $e \in \{-1, -2\}$) are rapidly increasing functions. When the spread is high, it is clearly pressed downward through inhibiting the positive

events (as indicated by small $f^{+1}(s)$ and $f^{+2}(s)$ values for large spreads), and exciting the negative events (large $f^{-1}(s)$ and $f^{-2}(s)$ values).

The two functions f^{-1} and f^{-2} are clearly saturating though saturation seems to appear faster on f^{-1} than on f^{-2}. It is important to recall that the estimation spreads larger than the chosen $\bar{S}$ is impossible, due to statistical limitations. In other words, spread values above the chosen $\bar{S}$ are extremely rare, leading to insufficient data for estimation.

Let us remark that the range of values attained by the functions associated with negative events is in line with the corresponding tick size and the average spread size of each asset (see Table 3.1). Indeed, large tick size corresponds to small average spread and small range of f^{-k} values. Specifically, the range increases from CAC40 Future (with the largest tick size) to NOKIA (with the smallest tick size).

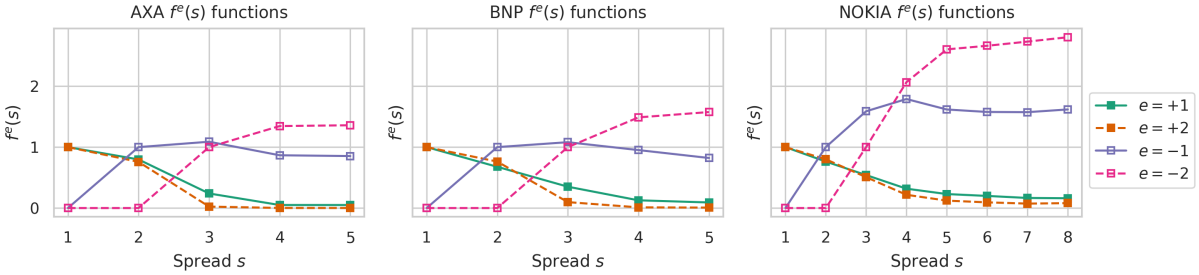


Figure 3.3 – Estimations of the $\{f^e(s)\}_{e\in\mathcal{E}}$ functions for AXA (left), BNP (middle), NOKIA (right), $\mathcal{E} = \{-2, -1, +1, +2\}$. As for CAC40 Future, the $\bar{S}$ is 2 and the K is 1, therefore $f^{-1}(1) = 0, f^{-1}(s) = 1$ if $s \geqslant 2$, $f^{+1}(1) = 1, f^{+1}(s) = 0.0104$ if $s \geqslant 2$. For each e, the first positive value of $f^e(S)$ is arbitrarily set to 1.

b. Estimation of Hawkes kernels $\{\varphi^{e,e'}(t)\}_{e,e'\in\mathcal{E}}$

First, let us point out that our estimation procedure essentially leads to positive-valued kernels. In practice, the MLE procedure described in Section 3.3.2 is able to detect inhibition behavior (i.e., significantly negative-valued kernels) when it is present in the signal. However, in our estimated kernels, even though some kernels might exhibit minor negative values, these values are very small and not significant. Of course, this does not mean that there is no inhibition behavior in the spread counting process. In fact, the inhibition behavior is extremely strong but it is effectively managed for each component S_t^e, through the multiplicative term $f^e(S_{t-})$. This is the main reason why we initially introduced them in our model (with "hard" inhibition of some components to prevent negative spread values).

Comparison of kernel integrated quantities

Let emphasize that, the introduction of multiplicative terms in the "State Dependent" Hawkes model (i.e., the fact that our model is not a standard Hawkes model due to the role of the state variable S_t) prevents us from interpreting the L^1 norm of the Hawkes kernels in the usual way. Indeed, in a classical Hawkes model, one generally compares the different values of the L^1 norms $\{\|\varphi^{e,e'}(t)\|_1\}_{e,e'\in\mathcal{E}}$ and uses the classical population-based interpretation of a Hawkes model. In this case, $\|\varphi^{e,e'}(t)\|_1$ represents the average number of events of type e' "directly" generated by an event of type e. In our model, this interpretation is not possible. Not only is it conditioned on a spread

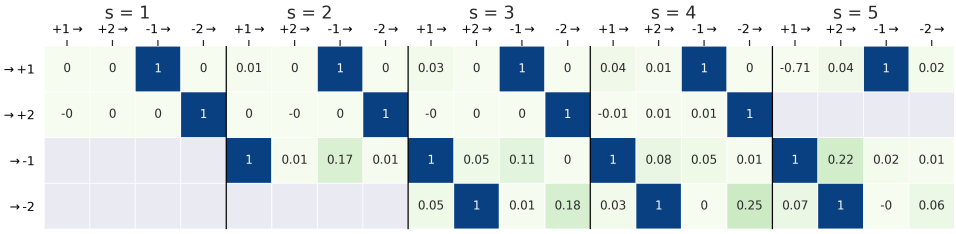


Figure 3.4 – Relative kernel integrated quantities $\tilde{I}_{s,e}(e')$ for AXA. For each stock, for each value of s, an image is displayed showing $\tilde{I}_{s,e}(e')$ (defined by (3.4.3)) as a function of e (vertical axis) and e' (horizontal axis).

value but also the comparison between two norm values $\|\varphi^{e_1,e'}(t)\|_1$ and $\|\varphi^{e_2,e'}(t)\|_1$ for two different components $e_1 \neq e_2$ does not make sense.

However, we can still compare the influence of all past events of a given type e' on the occurrence of an event of type e. Indeed, let us consider a spread jump of size e at time t_k. This means the spread jumps from a value s (at time t_k-) to $s+e$ (at time t_k+), where $s = S_{t_k-}$. Then one could study the influence of each endogenous term in the sum (within the brackets) in (3.2.2), by comparing the relative values $\int_0^{t_k-} \varphi^{e,e'}(t-u)\, dS_u^{e'}$ for different e' in order to understand which event type e' has the most significant influence on the occurrence of the jump of size e at time t_k. For that purpose, let us introduce the quantity

$$I_{s,e,t_k}(e') := \int_0^{t_k-} \varphi^{e,e'}(t_k-u)\, dN_u^{e'}, \quad \forall e'$$

for a jump of size e occurring at time t_k which changes the spread value from s to $s+e$. We can then define the corresponding averaged values over all timestamps t_k where a jump of size e occurs at spread s:

$$\bar{I}_{s,e}(e') := \frac{\sum_{t_k : dS_{t_k}=e, S_{t_k-}=s} I_{s,e,t_k}(e')}{\#\{t_k : dS_{t_k}=e, S_{t_k-}=s\}}.$$

and finally the relative values

$$\tilde{I}_{s,e}(e') = \frac{\bar{I}_{s,e}(e')}{\sup_{e'} \bar{I}_{s,e}(e')} \tag{3.4.3}$$

Figure 3.C.1 displays this quantity for AXA, showing $\tilde{I}_{s,e}(e')$ for each value of s. In the image plot, the vertical axis is e and the horizontal axis is e'.[1] Similar plots for other stocks can be found in Appendix 3.C Figure 3.C.1. These figures reveal a common pattern among stocks: the occurrence of a jump of size e is essentially triggered by past occurrences of opposing jumps of size $-e$.

Comparison of the kernel shapes

Finally, estimation results show that the most energetic kernels are decreasing "slowly", resembling a power-law decay $t^{-\beta}$ with an exponent $\beta \simeq 1$. Such a power-law shape of the cross-excitation kernels is not surprising since this behavior with similar exponent values have been observed by various authors when modelling the market activity Bacry et al. (2015) or the dynamics of mid-price

1. Let us note that, for CAC40, since most of the time the spread is 1 tick $S = 1$ (the only other possible state is $S = 2$ which happens very rarely), this type of analysis is not relevant

Bacry et al. (2016). Figure 3.5 shows the contrariant kernels for AXA in a log-log plot. We see that they display a power-law behavior on 3 or 4 decades (depending on the kernel). (See Figure 3.C.2a, 3.C.2b, 3.C.2c for the other assets in Appendix 3.C.)

Moreover, most of them display a very clear bump around the time $t \simeq 0.2$ milliseconds. This is not very surprising as this phenomenon has already been revealed in several former works (Bacry et al., 2016; Rambaldi et al., 2017). It corresponds to an average latency of the market itself, which is the average time for an agent to effectively place an order, reacting to a change of the order book.

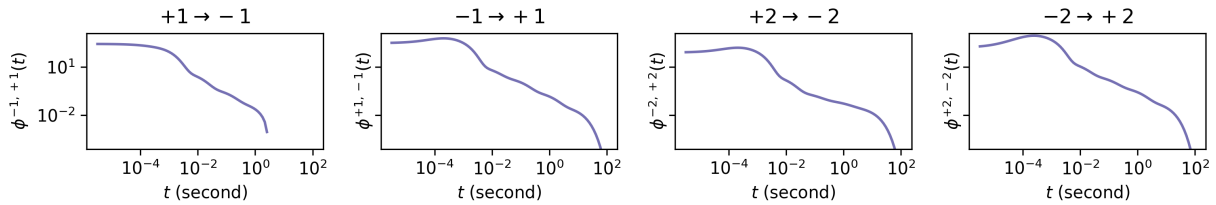


Figure 3.5 – Hawkes kernels for AXA. In this figure we present four contrariant Hawkes kernels (i.e., $\varphi^{e,e'}$ for $e = -e'$). Each kernel $\varphi^{e,e'}$ (labeled $e' \to e$ on the figure with $e, e' \in \mathcal{E}$) represents the influence of the past jumps of size e' on the occurrence probability of a future jumps of size e. Each kernel is represented as a sum of $L = 6$ exponentials, specifically $\varphi^{e,e'}(t)$, where $\varphi^{e,e'}(t) = \sum_{l=1}^{L} \alpha_l^{e,e'} \beta_l e^{-\beta_l t}$, where $\beta_l = \frac{1}{\tau_l}$ and τ_l takes values in the range $\{10^{-4}s, 10^{-3}s, \dots, 10^{1}s\}$. All the kernels are displayed on a log-log scale and show a power-law behavior on a large range of scales (3, 4 or even 5 decades)

3.4.4 Goodness-of-fit

In this section, we demonstrate that the previously constructed and estimated model is able to accurately capture diverse statistical properties of the spread process. We will study successively, the spread distribution itself (that has already been discussed in Section 3.4.2), the inter-event time distributions (i.e., time between change of spread values), the spread autocorrelation function and finally the auto-covariance function of the spread increment process.

a. Spread Distributions

Let us first compare the spread distributions derived from the real data with the distributions obtained from data simulated by our model, which was fitted to the real data. As discussed in Section 3.4.2, both calendar time and event time distributions are considered.

Following the same lines as in Section 3.4.2, Figure 3.6 displays calendar-time spread distribution, event-time distribution and spread jumps (dS) distribution for the AXA asset. The figures demonstrate that the model accurately replicates these distributions. Several works have explored the distribution of spread, but Fosset et al. (2020a) is one of the few that discuss it in detail. In their spread model (as shown in Equation (3.1.2)), the distribution of spread is geometric, given by:

$$\mathbb{P}(S \geqslant n) = \frac{1-\alpha_c}{1-\alpha}(1-r)r^{n-2} \qquad \text{for } n \geqslant 2$$

where $\alpha_c = 1 - \frac{\mu^+}{\mu^-}$ and $r \in (0,1)$ depends on α and β. However, since $\mathbb{P}(S = n)$ is a decreasing

function for $n \geqslant 2$, the model Eq. (3.1.2) can only produce spread distributions that peak at $S = 1$ or $S = 2$. As shown in Figure 3.6 and 3.C.3, this Characteristic is not consistent with real data.

It is also important to note that the SDSH model is not limited to reproducing the spread distribution, but is a more comprehensive framework which can capture a wider range of spread dynamics.

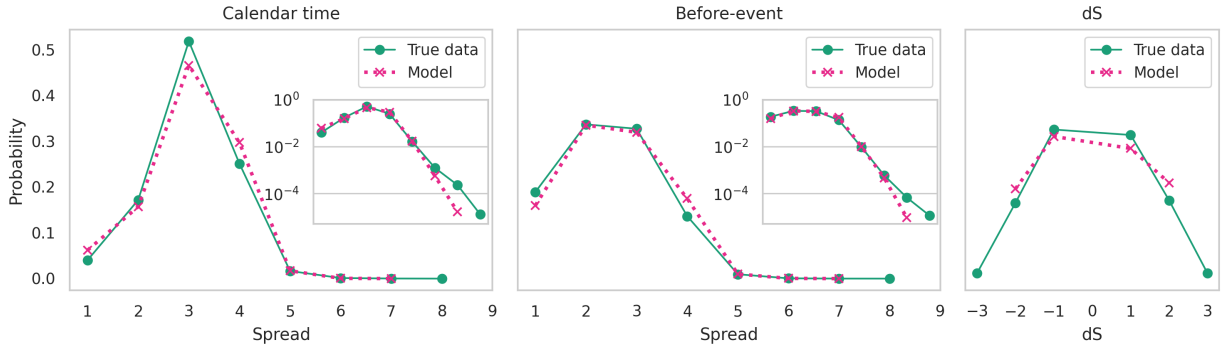


Figure 3.6 – Spread distribution, comparison between true data and the the data obtained through simulation for AXA. The left-hand figure is the Calendar time distributions, the middle figure is the event time distributions, and the right-hand figure is the distributions of spread jumps size.

b. Inter-event time distributions

Let $\{t_n\}_n$ be the successive times at which the spread process S jumps (i.e., the spread changes). We define the inter-event times $\{\Delta t_n\}_n$ by

$$\Delta t_n = t_n - t_{n-1}.$$

The first plot on the left of Figure 3.7a displays the empirical unconditional distribution of inter-event times for both the AXA true data and simulated data using our model (fitted with AXA data). The two other plots on the right hand-side of this latter plot show some conditional inter-event time distributions. More precisely we define the set

$$\{\Delta t_{S_1 \to S_2}\} := \{\Delta t_i \mid S(t_i+) = S_1,\ S(t_{i+1}+) = S_2\}, \tag{3.4.4}$$

where $S(t_i+)$ is the value of the spread immediately after the jump at time t_i. The middle plot in Figure 3.7a shows the distribution of $\{\Delta t_{1\to 3}\}$, while the right one shows the distribution of $\{\Delta t_{3\to 2}\}$.

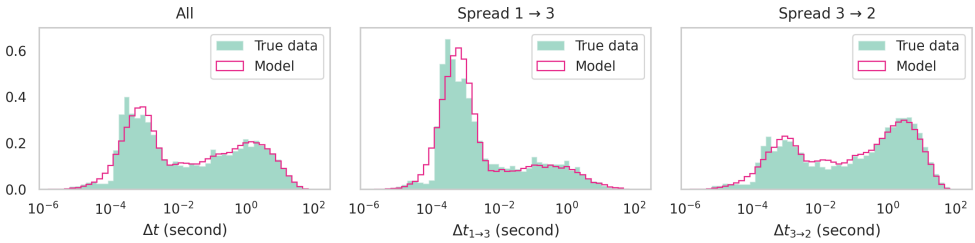
We see that the model performs extremely well in reproducing the unconditional and conditional inter-event time distributions. Figure 3.7b displays the qq-plots of the true distributions versus the model distributions. Their linear behavior show how well the model fits the true distributions.

We invite reader to look at the results obtained for BNP, NOKIA or CAC40 Future in Appendix 3.C (Figures 3.C.4, 3.C.5 and 3.C.6).

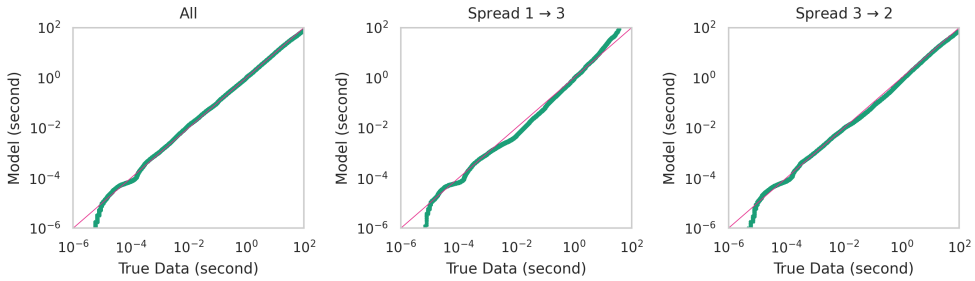
c. Spread Autocorrelation

In this section, we study the auto-correlation function of the spread for true data and assess how well the SDSH model is able to reproduce it.

The autocorrelation function of the spread on true data could a priori be estimated using straightforward quadratic covariations on all the available data. However such an estimation is prone to



(a)



(b)

Figure 3.7 – AXA Inter-event time unconditional and conditional distributions. Comparison between AXA true data and data obtained through simulation of our model (fitted on AXA data) (a) AXA true distributions versus model distributions for unconditional distribution or conditional distributions (see (3.4.4)). The x-axis is on log scale. (b) Corresponding qq-plots in log-log scales.

highly biased results. Indeed, it is well known that the order book dynamics (and subsequently the spread dynamics) is subject to long range correlations and strong intraday seasonal effects (Bouchaud et al., 2009; Fall et al., 2021; Groß-KlußMann and Hautsch, 2013). To avoid these biases, it is important to account for the intraday seasonality. A common approach is to limit the computation of the quadratic covariations to certain time slots every day, by assuming that the seasonal effects between different days are less affecting.

In our case, we use 15min time slots on the real data. More precisely we use eight 15min slots between 10am and 12pm each day. On the other hand, for the simulated data, we keep the 2 hour time-slots since there is no seasonality in the model. Figure 3.8 shows these estimations for AXA and additional plots for BNP, NOKIA and CAC40 are available in Appendix 3.C Figure 3.C.7. Each plot displays the autocorrelation functions for both true data and simulated data, shown in linear-linear and log-log scales. Once more, the fits are amazingly good. The model succeeds in reproducing the autocorrelation function of the spread with a very good accuracy.

d. Autocovariance of spread increments

In this section we focus on the autocovariance function of the spread increments. Let us first give the exact definition.

Let us define the infinitesimal covariance of the infinitesimal measure dS_t as $Cov(dS_t, dS_{t'})$, which is assumed to be stationary and dependent only on $(t' - t)$. We refer to this covariance as $g(t' - t)$,

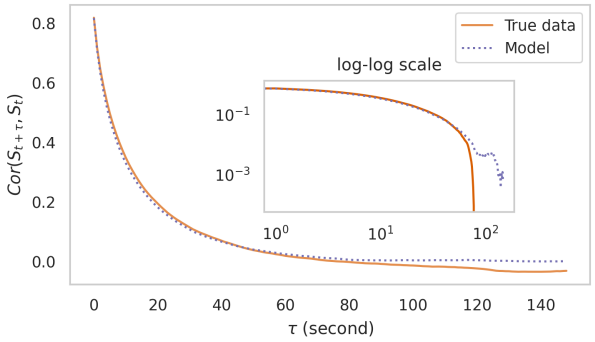


Figure 3.8 – Auto-correlation function of the spread for AXA. True data versus model-simulated data. Each plot corresponds to a different asset. Quadratic variations are used for estimation. Eight 15min time-slots (between 10am and 12pm) are used everyday for true data in order to avoid intraday seasonal effects. 2-hour slots are used for model-simulated data.

such that:

$$\begin{aligned} g(t'-t)dtdt' &= Cov(dS_t, dS_{t'}) \\ &= \mathbf{E}[dS_t\, dS_{t'}] - \mathbf{E}[dS_t]\mathbf{E}[dS_{t'}], \end{aligned}$$

Then, one gets, $\forall \delta > 0$ and $\forall \tau > 0$,

$$\begin{aligned} Cov(S_{t+\delta} - S_t, S_{t+\delta+\tau} - S_{t+\tau}) &= \mathbf{E}[\int_t^{t+\delta} dS_u \int_{t+\tau}^{t+\tau+\delta} dS_v] - \mathbf{E}[\int_t^{t+\delta} dS_u]\mathbf{E}[\int_{t+\tau}^{t+\tau+\delta} dS_v] \\ &= \int_t^{t+\delta}\int_{t+\tau}^{t+\tau+\delta} (\mathbf{E}[dS_u\, dS_v] - \mathbf{E}[dS_u]\mathbf{E}[dS_v]) \\ &= \int_t^{t+\delta}\int_{t+\tau}^{t+\tau+\delta} g(v-u)\, du\, dv \\ &= \delta^2 \int_0^1\int_0^1 g(\tau + \delta(v-u))\, du\, dv \end{aligned}$$

It is thus natural to introduce a normalized quantity, which represents the relative covariance of spread increment during δ seconds with lag τ seconds:

$$ACV(\delta, \tau) := \frac{1}{\delta^2} Cov(S_{t+\delta} - S_t, S_{t+\delta+\tau} - S_{t+\tau}) \tag{3.4.5}$$

Let us point out that the function $g(\tau)$ corresponds to the infinitesimal covariance function of a stationary process, it should thus decreases to 0 when the lag τ goes to infinity. It seems reasonable to assume that it does so in a "regular way". This assumption can be expressed mathematically as follows: g is differentiable and there exist constant $\epsilon > 0$ and $c > 0$ such that

$$|g'(\tau)| < c\tau^{-\epsilon}, \quad \forall \tau > 0$$

Under this assumption, we have, for any $\tau > \delta > \delta' > 0$, that

$$\begin{aligned}
|ACV(\delta,\tau) - ACV(\delta',\tau)| &= \left| \int_0^1 \int_0^1 (g(\tau + \delta(v-u)) - g(\tau + \delta'(v-u)))\, du\, dv \right| \\
&\leqslant (\delta - \delta') \max_{|y| \in [0, \max(\delta,\delta')]} |g'(\tau + y)| \int_0^1 \int_0^1 |v-u|\, du\, dv \\
&\leqslant \frac{1}{3}(\delta - \delta') \max_{|y| \in [0, \max(\delta,\delta')]} |g'(\tau + y)| \\
&\leqslant \frac{c(\delta - \delta')}{3(\tau - \delta)^{\epsilon}}
\end{aligned}$$

Consequently, the covariance function $ACV(\delta,\tau)$ can be estimated independently of the value of δ as long as the lag τ is large enough compared to the value of δ. In practice, to avoid estimating ACV for all possible values of δ and τ, one can fix a very small value for δ and then estimate the ACV function on a range of τ that satisfies $\tau \gg \delta$. However, as we will see, if δ is chosen to be extremely small compared to τ, the estimation can become very noisy due to the high-frequency fluctuations. Therefore to get a smooth estimation, one needs to choose a value for δ that is much smaller that τ but not so small that it introduces excessive noise.

Figure 3.9 not only visually reinforces this discussion but demonstrates that the model perfectly reproduces all the statistical features of the true data (of the AXA asset). We choose to represent $-ACV$ instead of ACV since due to the mean reversion property of the spread, we naturally expect ACV to be mainly negative. The inset in the figure shows $-ACV(\delta,\tau)$ as a function of τ in a log-log scale for a fixed $\delta = 0.1s$. One sees that if δ is too small compared to τ, the result gets extremely noisy. Moreover it is clear from the plot that the autocovariance function computed using the model simulated data reproduces very well the behavior of the one computed using the true data.

The main plot displays the different estimations of the $ACV(\delta,\tau)$ as a function of τ for different values of δ, indicated in the legend on the right hand-side. Following what we just said, we limited for each δ the estimation of $ACV(\delta,\tau)$ for values of τ on a range so that δ is small compared to τ but not too small. This approach results in all the estimation curves practically overlapping, revealing a smooth curve for the auto-covariance function spanning nearly 7 decades of τ values.

This curve is close to be linear, indicating that the auto-covariance function is close to be power-law. One can see the slight latency bump around $\tau = 200\mu s$, previously discussed in Section b for Figures 3.5 and 3.C.2. However, the most impressive is the way the estimation computed using the model-simulated data fits the estimation on true data on the whole range of scales. The fit is very accurate.

Last but not least, it's worth noting that the same results hold true when considering the other assets. This is illustrated in Figure 3.C.8. The remarkable fit between the model and the data remains consistent across the assets, and the auto-covariance curves exhibit a similar pattern. This consistent behavior across multiple assets suggests the possibility of a stylized fact present in the empirical processes of spreads.

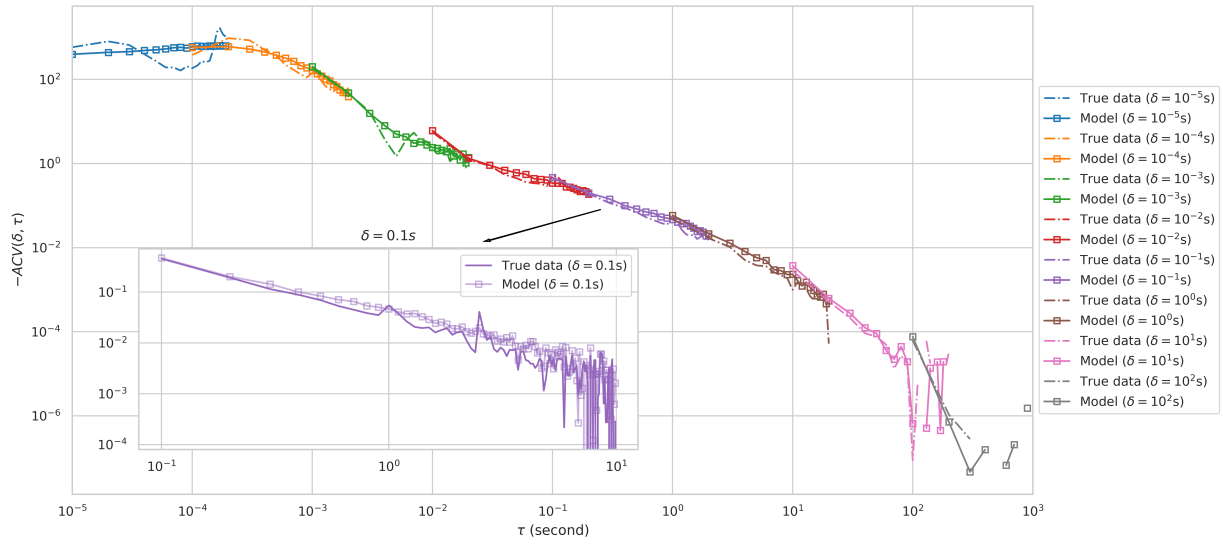


Figure 3.9 – The $-ACV(\delta,\tau)$ functions for different values of δ as a function of τ using a log-log scale both for using the AXA true data and the model-simulated data (fitted on AXA true data). As expected (see discussion) the curves for different δ's fall one onto the over letting discover a power-law behavior with a slight bump close to the latency time scale $\simeq 200\mu s$ (see Section b). The inset shows the particular curve $ACV(\delta,\tau)$ for $\delta = 0.1s$. It clearly confirms the fact that if τ is too far to δ the estimation gets very noisy.

3.5 - Illustration on using SDSH model for spread forecasting

The goal of this section is to show that the SDSH model is a good candidate for high-frequency spread forecasting. However, it is important to note that spread forecasting is a difficult task that deserves much more dedicated research and comparisons with state of the art methods. It is clearly beyond the scope of our current study. In this section, our aims are more modest. We just intend to give some very preliminary results that allow us to glimpse the possibilities of such an application.

The issue we address is the prediction of the spread value at a specific time horizon Δ in the very near future (i.e., $\Delta < 1$ min). According to Figure 3.9, a one-minute time window should be sufficient to capture the direct impact of past events on future spread values. Taking into account only the direct impact should be enough to capture most of the dynamics and to give an estimation of the performance achievable by the SDSH model. Since we lack an explicit expression for the spread distribution, we rely on Monte Carlo simulations to estimate the expected spread value at the given time horizon. To elaborate further:

At time t_0, the spread point process during the time interval $[t_0-60, t_0]$ can be summarized by either $\{S_u\}_{t_0-60<u<t_0}$ or equivalently by S_{t_0-60} along with all the events $\{(t_i, e_i), t_0 - 60 \leqslant t_i < t_0\}_{i=1,\dots,n}$ occurring within the time-interval $[t_0 - 60, t_0]$. Thus, in order to estimate $\mathbb{E}\big[S_{t_0+\Delta} \,|\, S_u, u \in [t_0 - 60, t_0]\big]$, which represents the conditional mean at time-horizon Δ, we simulate 100 processes from t_0 to $t_0 + \Delta$ with the intensity function at time $t \in [t_0, t_0 + \Delta]$ defined as

$$\lambda_t^e = f^e(S_{t-})\big(\tilde{\mu}^e(t) + \sum_{e' \in \mathcal{E}} \int_{t_0}^{t-} \varphi^{e,e'}(t-u) dS_u^{e'}\big) \tag{3.5.1}$$

where the baseline $\tilde{\mu}^e(t)$ is a function on $t \in [t_0, t_0 + \Delta]$ defined by

$$\tilde{\mu}^e(t) = \mu^e + \sum_{e' \in \mathcal{E}} \sum_{i=1}^{n} 1_{e_i = e'} \varphi^{e,e'}(t - t_i) \; . \tag{3.5.2}$$

Before illustrating the performance of such an estimator, let us briefly describe an alternative model, previously introduced in Groß-KlußMann and Hautsch (2013), that we will use as a benchmark for our numerical tests.

Introducing a benchmark: the Autoregressive Conditional Double Poisson (ACDP) model In Groß-KlußMann and Hautsch (2013), the authors introduce various spread prediction models which are all based on the use of autoregressive conditional Poisson model. As our focus is on high-frequency prediction, we will consider only the so-called ACDP model and not its long-memory version (i.e., the LMACP, the Long Memory Autoregressive Conditional Poisson model). Once again, it is important to emphasize that the development of an optimized spread forecasting procedure based on the SDSH model, as well as conducting a thorough comparison with state-of-the-art methods, falls beyond the scope of this study. A forthcoming work will be specifically dedicated to these topics.

Within the ACDP framework, the spread process $S_k, k \in \mathbb{Z}$ is a discrete time series, consisting of spread values subsampled every $\Delta = 30s$ (as shown below, we explore the effects of varying this sub-sampling interval which plays also the role of the time-horizon between 3 seconds and 30 seconds):

$$\begin{aligned} \lambda_k &= c + \alpha S'_{k-1} + \beta \lambda_{k-1} \\ S'_k | \mathcal{F}_{k-1} &\sim \mathcal{DP}(\lambda_k, \gamma) \end{aligned} \tag{3.5.3}$$

where $S'_k = S_k - 1 \in \mathbb{N}$, and $\mathcal{DP}(\lambda_k, \gamma)$ stands for the Double Poisson distribution defined by

$$\mathbb{P}(S'_k = n | \lambda_k, \gamma) = c(\gamma, \lambda_k) \gamma^{1/2} e^{-\gamma \lambda_k} \big(\frac{e^{-n} n^n}{n!}\big) \big(\frac{e \lambda_k}{n}\big)^{\gamma n}$$

Under this model, $\mathbb{E}[S'_k \,|\, \mathcal{F}_{k-1}] = \lambda_k$.

As shown in Groß-KlußMann and Hautsch (2013), the log-likelihood function of ACDP model reads:

$$\log \mathcal{L}(c, \alpha, \beta, \gamma | S'_{[1:N]}) = \sum_{k=1}^{T} \Big(\frac{1}{2} \log(\gamma) - \gamma \lambda_k + S'_k (\log S'_k - 1) - \log(S'_k!) + \gamma S'_k \big(1 + \log(\frac{\lambda_k}{S'_k})\big) \Big)$$

where $\lambda_t = (1 - \beta B)^{-1}(c + \alpha B(S'_t))$ with $B(S'_t) = S'_{t-1}$ being the backshift operator. When $\beta < 1$:

$$\lambda_t = \sum_{i=0}^{\infty} \beta^i B^i (c + \alpha B(S'_t))$$

In practice, one estimates λ_t based on a truncation of the infinite sum. Specifically it is estimated as $\lambda_t = \sum_{i=0}^{N} \beta^i B^i (c + \alpha B(S'_t))$, where N is an hyper parameter of the method. In this work, we set N to be 60. In fact, after experimenting with different values of N, we found that the results remain consistent as long as N is greater than 10.

Numerical results Let us present the numerical results on the relative performances of SDSH and ACDP based methods. We are going to compare three different predictors for time-horizons varying from $\Delta = 3s$ up to $\Delta = 30s$.

- **SDSH predictor**:
  - For each stock, the SDSH model is calibrated on training data which consists of 30-day spread dynamics within a 2-hour window each day (from 10am to 12pm) for avoiding strong seasonal effects.
  - No re-estimation of the model parameters is made along running the test
  - At each time, forecasting is made using the previous 1 minute data following Equations (3.5.1) and (3.5.2)
- **ACDP predictor**:
  - For each stock, the ACDP model is calibrated using a time rolling window of 1-hour of spread data subsampled every Δ seconds.
  - Parameters are re-estimated every 10 minutes to ensure that the parameters remain up-to-date.
  - The model employs a one-step ahead prediction strategy using the most recent estimated model for forecasting.
- **Last predictor**: A straightforward benchmark predictor involves assuming that S_t follows a martingale, meaning that the prediction for the spread at time $t_0 + \Delta$ is the last observed value of the spread, denoted as $\hat{S}_{t_0+\Delta} = S_{t_0}$.

All methods are evaluated over a test period lasting 50 consecutive days, following the initial 30-day training period for the SDSH model. The evaluation takes place specifically from 11am to 12pm every day. This time-frame is chosen to accommodate the ACDP method, as it requires a minimum of one hour to calibrate the model before generating predictions. Therefore, we cannot make predictions before 11am.

We evaluate these three predictors on three stocks (AXA, BNP Paribas and Nokia) as well as the CAC40 index future. The performance comparison is presented in Table 3.3.

As table 3.3 shows, the Last predictor is always worse than the SDSH predictor though it outperforms the ACDP predictor sometimes at the highest frequency (e.g., $\Delta = 3s$, for AXA and NOKIA). Moreover the SDSH predictor outperforms most of the time the ACDP predictor (and systematically for the highest frequency $\Delta = 3s$). These results are very encouraging in terms of the predictive performance of the SDSH model. A detailed comparison with several state of the art predictors and for a wider range of time horizons will be addressed in a forthcoming work.

3.6 - Conclusion

In this study, we introduced a State Dependent Hawkes process (SDSH) for modelling bid-ask spread fluctuations, which generalizes the spread models presented in Zheng et al. (2014) and Fosset et al. (2020a). Our model is a 2K-variate Hawkes process which can accommodate different jump sizes (ranging from 1 to K). In order to account for the current spread value S_t, we introduced a spread-dependent term $f^e(S_t)$ (where e denotes an event type) and multiplied the classical Hawkes intensity by this term. We chose to use the sum of exponential kernels to benefit from the Markovian properties of the model. Notably, we demonstrated the ergodicity property in a particular case, indicating that the spread process converges to a stationary distribution over time.

Δ	3s	6s	12s	30s
Last	0.408	0.587	0.784	1.082
ACDP	0.514	0.520	0.565	0.808
SDSH	**0.363**	**0.478**	**0.561**	**0.648**

(a) AXA

Δ	3s	6s	12s	30s
Last	0.833	1.177	1.487	1.818
ACDP	0.737	**0.851**	0.971	1.320
SDSH	**0.692**	0.865	**0.964**	**1.065**

(b) BNP

Δ	3s	6s	12s	30s
Last	0.388	0.609	0.904	1.340
ACDP	0.508	0.609	**0.744**	**1.031**
SDSH	**0.361**	**0.543**	0.766	1.081

(c) NOKIA

Δ	3s	6s	12s	30s
Last	0.370	0.421	0.445	0.457
ACDP	0.283	0.258	0.256	0.267
SDSH	**0.230**	**0.240**	**0.244**	**0.245**

(d) CAC40 Index Future

Table 3.3 – Mean Square Error (MSE) of the 3 different predictors, Last, ACDP and SDSH for different time sub-samplings Δ (which plays also the role of the time-horizon).

Then we calibrated our SDSH model using high-frequency data obtained from CAC40 Euronext Market, including three stocks (AXA, BNP, Nokia) and the CAC40 Future index. We examined the estimated spread-dependent term f, as well as kernel functions to better understand how they affect the spread dynamics. Our analysis revealed that the estimated $f^e(\cdot)$ is decreasing when e is an event which increased the spread, while for at least one downward event e', $f^{e'}(\cdot)$ is increasing. Such f^e is able to press the spread down when its value is high by exciting more downward events. In terms of kernel functions, our estimation results showed that most of them tend to decrease very slowly, resembling a power-law kernel function.

We studied the ability of this model to capture various spread statistics. We found that our model very successfully replicates the spread distributions, measured by both calendar time and event time. We also observed that the model accurately reproduces other important statistical features, including the distributions of inter-event times, spread autocorrelations as well as spread increments autocovariance.

Finally we demonstrated the effectiveness of the SDSH model in predicting the spread across different sample window sizes. Our results suggest that the SDSH model is a reliable and robust choice for predicting the spread.

APPENDIX

3.A - Proof of V-uniform ergodicity

We restrict our model to the case where $K = 1$ and $L = 1$. The model then is written as follows:

$$\lambda^e(t) = f^e(S_{t-})(\mu^e + \sum_{e' \in \mathcal{E}} \int_0^t \varphi^{e,e'}(t-s) dS_s^{e'})$$

where $\mathcal{E} = \{+1, -1\} =: \{+, -\}$ and $\varphi^{e,e'}(t) = \alpha^{e,e'} \beta e^{-\beta t}$

This section is devoted to the prove of the Proposition 3.2 of Section 3.2.3. From now, we replace event + by 1 and − by 2. To avoid the ambiguity of notation, in the following part, we replace the superscripts by subscripts. Under the new notations, we replicate the proposition here:

Proposition. *Let us consider the model given just above and* $X = (X_{lm})_{l,m \in \{1,2\}}$. *Assume that the following conditions are satisfied*

$$\boxed{\begin{array}{l} f_2(1) = 0 \\ f_2(S) \geqslant \gamma S \text{ for some } \gamma > 0 \text{ when } S \geqslant 2 \\ \sup_S \{f_1(S)\}(\alpha_{12} + \alpha_{11}) < 1 \end{array}} \tag{A3}$$

then the process (S_t, X_t) *is a V-uniformly ergodic Markov process.*

Remark 3.1. *Let us note that in this model, the last condition* $\sup_S \{f_1(S)\}(\alpha_{12} + \alpha_{11}) < 1$ *is equivalent to* $(\alpha_{12} + \alpha_{11}) < 1$ *and* $\sup_S \{f_1(S)\} <= 1$.

Lyapunov function

As $X_{lm}(t) = \int_0^t \alpha_{lm} e^{-\beta(t-s)} dN_s^m$,

$$dX_{lm}(t) = -\beta X_{lm}(t) dt + \alpha_{lm} dN_t^m$$

Function $V_{lm}(X_{lm})$

$$V_{lm}(X_{lm}) = X_{lm}$$

The infinitesimal generator $\mathcal{L}$ of X_{lm} on V_{lm}

$$\begin{aligned}\mathcal{L}V_{lm}(X) &= \alpha_{lm}\lambda_m - \beta X_{lm} \\ &= -\beta X_{lm} + \alpha_{lm} f_m(S)\mu_m + \alpha_{lm} f_m(S)(\sum_n \beta X_{mn}) \\ &= -\beta X_{lm} + \alpha_{lm}\mu_m f_m(S) + \alpha_{lm}\beta f_m(S)(X_{m1} + X_{m2})\end{aligned}$$

Function $V_S(S)$

$$V_2(S) = S$$

As $dS_t = dN_t^1 - dN_t^2$, the infinitesimal generator $\mathcal{L}$ of S on V_S:

$$\begin{aligned}\mathcal{L}V_2(S) &= \lambda_1 - \lambda_2 \\ &= f_1(S)\mu_1 + \beta f_1(S)X_{11} + \beta f_1(S)X_{12} - f_2(S)\mu_2 - \beta f_2(S)X_{21} - \beta f_2(S)X_{22}\end{aligned}$$

Function V on (X, S) Now we consider a function V on (X, S)

$$V(X, S) = \sum_{l,m\in\{1,2\}} \eta_{lm} X_{lm} + \eta S$$

where $\eta_{lm}, \eta > 0$.

Then the infinitesimal generator $\mathcal{L}$ of (X, S) on V is:

$$\begin{aligned}\mathcal{L}V(X, S) = &-\eta_{11}\beta X_{11} + \eta_{11}\alpha_{11}\mu_1 f_1(S) + \eta_{11}\alpha_{11}\beta f_1(S)(X_{11} + X_{12}) \\ &-\eta_{12}\beta X_{12} + \eta_{12}\alpha_{12}\mu_2 f_2(S) + \eta_{12}\alpha_{12}\beta f_2(S)(X_{21} + X_{22}) \\ &-\eta_{21}\beta X_{21} + \eta_{21}\alpha_{21}\mu_1 f_1(S) + \eta_{21}\alpha_{21}\beta f_1(S)(X_{11} + X_{12}) \\ &-\eta_{22}\beta X_{22} + \eta_{22}\alpha_{22}\mu_2 f_2(S) + \eta_{22}\alpha_{22}\beta f_2(S)(X_{21} + X_{22}) \\ &+\eta f_1(S)\mu_1 + \eta\beta f_1(S)X_{11} + \eta\beta f_1(S)X_{12} \\ &-\eta f_2(S)\mu_2 - \eta\beta f_2(S)X_{21} - \eta\beta f_2(S)X_{22}\end{aligned}$$

Before giving the proof of Proposition 3.2, we should mention the following theorem. See Theorem 5.2 in Down et al. (1995) and 2.5.2 in Abergel and Jedidi (2013).

Theorem 3.1. *For a ψ-irreducible, aperiodic Markov process X, if the following drift condition $(\mathcal{D})$ holds, then X is V-uniformly ergodic.*

$$(\mathcal{D})\ \textit{For some } \rho, b > 0 \textit{ and a coercive function } V \geqslant 1$$

$$\mathcal{L}V \leqslant -\rho V + b \tag{3.A.1}$$

Proof of Theorem 3.2. Suppose that we have already the following conditions:

$$(\mathcal{H}) = \begin{cases} \eta_{12}\alpha_{12} + \eta_{22}\alpha_{22} < \eta & (\mathrm{H}_1) \\ \eta_{11}\alpha_{11} + \eta_{21}\alpha_{21} + \eta < \eta_{11} & (\mathrm{H}_2) \\ \eta_{11}\alpha_{11} + \eta_{21}\alpha_{21} + \eta < \eta_{12} & (\mathrm{H}_3) \end{cases}$$

To proceed, we will split $\mathcal{L}V(X,S)$ into four distinct parts and show their individual upper bounds.

$$\mathcal{L}V(X,S) = I_1 + I_2 + I_3 + I_4$$

where

$$\begin{aligned}
I_1 &= (1) + (2) \\
&= -\eta_{11}\beta X_{11} - \eta_{12}\beta X_{12} \\
&+ \eta_{11}\alpha_{11}\beta f_1(S)(X_{11} + X_{12}) + \eta_{21}\alpha_{21}\beta f_1(S)(X_{11} + X_{12}) + \eta\beta f_1(S)(X_{11} + X_{12}) \\
&< (-\eta_{11} + \eta_{11}\alpha_{11} + \eta_{21}\alpha_{21} + \eta)\beta X_{11} + (-\eta_{12} + \eta_{11}\alpha_{11} + \eta_{21}\alpha_{21} + \eta)\beta X_{12} \\
&< -\epsilon_1\beta X_{11} - \epsilon_2\beta X_{12}
\end{aligned}$$

this last inequality is directly derived by conditions (H_2) and (H_3).

$$\begin{aligned}
I_2 &= (3) + (4) \\
&= -\eta_{21}\beta X_{21} - \eta_{22}\beta X_{22} \\
&+ \eta_{12}\alpha_{12}\beta f_2(S)(X_{21} + X_{22}) + \eta_{22}\alpha_{22}\beta f_2(S)(X_{21} + X_{22}) - \eta\beta f_2(S)(X_{21} + X_{22}) \\
&= -\eta_{21}\beta X_{21} - \eta_{22}\beta X_{22} + (\eta_{12}\alpha_{12} + \eta_{22}\alpha_{22} - \eta)\beta f_2(S)(X_{21} + X_{22}) \\
&\overset{(H_1)}{<} -\eta_{21}\beta X_{21} - \eta_{22}\beta X_{22}
\end{aligned}$$

Now for the other two terms I_3 and I_4:

$$\begin{aligned}
I_3 &= (5a) \\
&= \eta_{11}\alpha_{11}\mu_1 f_1(S) + \eta_{21}\alpha_{21}\mu_1 f_1(S) + \eta\mu_1 f_1(S) < \eta_{11}\alpha_{11}\mu_1 + \eta_{21}\alpha_{21}\mu_1 + \eta\mu_1 =: C_1
\end{aligned}$$

$$\begin{aligned}
I_4 &= (5b) \\
&= \eta_{12}\alpha_{12}\mu_2 f_2(S) + \eta_{22}\alpha_{22}\mu_2 f_2(S) - \eta f_2(S)\mu_2 = (\eta_{12}\alpha_{12} + \eta_{22}\alpha_{22} - \eta) f_2(S)\mu_2 \\
&< -\epsilon_0\mu_2 f_2(S) \overset{(A_2)}{<} \epsilon_0\mu_2\gamma - \epsilon_0\mu_2\gamma S =: C_2 - -\epsilon_0\mu_2\gamma S
\end{aligned}$$

where

- $0 < \epsilon_0 < \eta - \eta_{12}\alpha_{12} - \eta_{22}\alpha_{22}$
- $0 < \epsilon_1 < \eta_{11} - (\eta_{11}\alpha_{11} + \eta_{21}\alpha_{21} + \eta)$
- $0 < \epsilon_2 < \eta_{12} - (\eta_{11}\alpha_{11} + \eta_{21}\alpha_{21} + \eta)$

Therefore

$$\begin{aligned}
\mathcal{L}V(X,S) &= I_1 + I_2 + I_3 + I_4 \\
&< -\epsilon_1\beta X_{11} - \epsilon_2\beta X_{12} - \eta_{21}\beta X_{21} - \eta_{22}\beta X_{22} + C_1 - \epsilon_0\mu_2 f_2(S) \\
&< -\epsilon_1\beta X_{11} - \epsilon_2\beta X_{12} - \eta_{21}\beta X_{21} - \eta_{22}\beta X_{22} + C_1 + C_2 - \epsilon_0\mu_2\gamma S \\
&< -\rho(\eta_{11}X_{11} + \eta_{12}X_{12} + \eta_{21}X_{21} + \eta_{22}X_{22} + \eta S) + C \\
&= -\rho V(X,S) + C
\end{aligned}$$

where $\rho = \min\{\frac{\epsilon_1}{\eta_{11}}, \frac{\epsilon_2}{\eta_{12}}, 1, \frac{\epsilon_0\mu_2\gamma}{\eta}\}, \beta > 0$ and $C = C_1 + C_2$

Now we only need to find some $\eta_{lm}, \eta > 0$ satisfying the hypothesis $(\mathcal{H})$ to finish this proof.

As $\alpha_{11}+\alpha_{12}<1$, $\dfrac{\alpha_{11}}{1-\alpha_{12}}<1<\dfrac{1-\alpha_{11}}{\alpha_{12}}$. We note $\delta=\dfrac{1}{2}(\dfrac{1-\alpha_{11}}{\alpha_{12}}-1)$. Then the following values for η_{lm},η

$$\begin{cases}\eta_{11}=1,\eta_{12}=\dfrac{1-\alpha_{11}}{\alpha_{12}}-\delta>1=\eta^{11}\\ \eta_{21}=\delta\dfrac{\alpha_{12}}{4\alpha_{21}},\eta_{22}=\delta\dfrac{\alpha_{12}}{4\alpha_{22}}\\ \eta=1-\alpha_{11}-\dfrac{1}{2}\delta\alpha_{12}\end{cases}\tag{3.A.2}$$

satisfy the condition $(\mathcal{H})$. □

Remark 3.2. *For a more complicated version of our model:*

$$\lambda^e(t)=f^e(S_{t-})(\mu^e+\sum_{e'\in\mathcal{E}}\int_0^t\varphi^{e,e'}(t-s)dS_s^{e'})$$

where $\mathcal{E}=\{+1,+2,-1,-2\}$, $\varphi^{e,e'}(t)=\alpha^{e,e'}\beta e^{-\beta t}$ *(exponential kernels). Using the same proof, we can prove that under the following conditions* $(\mathcal{A}_1)$ *and* $(\mathcal{H}_1)$*, the* (X,S) *is a V-uniformly ergodic Markov process.*

$$(\mathcal{A}_1)=\begin{cases}f^{-1}(S)=0\ \textit{when}\ S=1\\ f^{-2}(S)=0\ \textit{when}\ S=1,2\\ \max(f^{-1}(S),f^{-2}(S))\geqslant\gamma S\ \textit{for some}\ \gamma>0\ \textit{when}\ S\geqslant 3\\ f^{+1}(S),f^{+2}(S)\leqslant 1\ \textit{for all}\ S\end{cases}$$

$$(\mathcal{H}_1)=\begin{cases}\sum_e\eta_{e,-1}\alpha^{e,-1}<\eta\\ \sum_e\eta_{e,-2}\alpha^{e,-2}<2\eta\\ \sum_e\eta_{e,+1}\alpha^{e,+1}+\eta<\eta_{+1,e'},\forall e'\in\mathcal{E}\\ \sum_e\eta_{e,+2}\alpha^{e,+2}+2\eta<\eta_{+2,e'},\forall e'\in\mathcal{E}\end{cases}$$

And the coercive function V *is* $V(X,S)=\sum_{e,e'\in\mathcal{E}}\eta_{e,e'}X_{e,e'}+\eta S$

3.B - Log-likelihood function of spread model

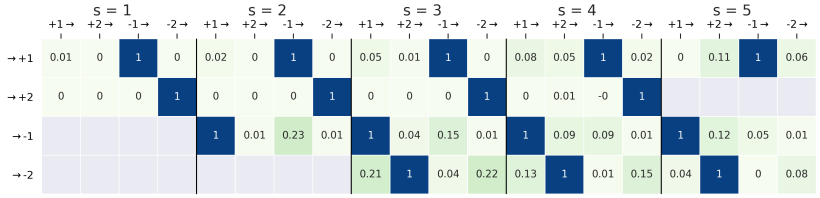
The log-likelihood function for the spread model is a function on μ, α and f. For brevity, we will only give the formula for the case where $L=1$ (only one decay). In order to distinguish from the notations that already exist, the log-likelihood function is denoted by $\mathbb{L}$. Consider a realization on $[0,T]$ and denote by $\{t_k^e\}$ the event times on S^e. The log-likelihood function is:

$$\begin{aligned}\mathbb{L}(\alpha,\mu,f)&=\sum_{e\in\mathcal{E}}(-\int_0^T\lambda^e(t)\,dt+\int_0^T\log\lambda^e(t)\,dS_t^e)\\&=\sum_{e\in\mathcal{E}}\sum_{k=1}^{S^e(T)}\log(\mu^e+\sum_{e'\in\mathcal{E}}\alpha^{ee'}\beta\int_0^{t_k^e}e^{-\beta(t_k^e-s)}\,dS_s^j)+\sum_{e\in\mathcal{E}}\sum_{k=1}^{S^e(T)}\log f^e(S_{t_k^e})\\&\quad-\sum_{e\in\mathcal{E}}\int_0^T(\mu^e+\sum_{e'\in\mathcal{E}}\alpha^{ee'}\beta\int_0^t e^{-\beta(t-s)}dS_s^{e'})f^e(S_t)\,dt\end{aligned}\tag{3.B.1}$$

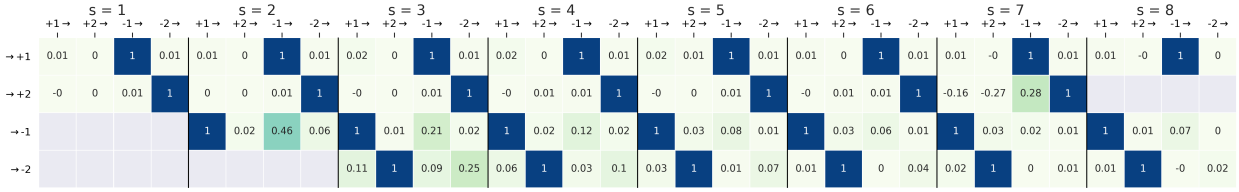
where $S^e(T)$ is the number of event e in $[0,T]$, and t_k^e is the timestamp where the kth event (type e) occurs.

3.C - More numerical results

In the main body of this paper, our empirical results are primarily centered on AXA. However, the optimal hyperparameters may vary when applied to different assets. In this section, we present similar empirical results for the other assets (BNP, Nokia and the futures contract of CAC40). These results serve to demonstrate the versatility and effectiveness of our model across a range of assets, thereby enhancing its performance and applicability.

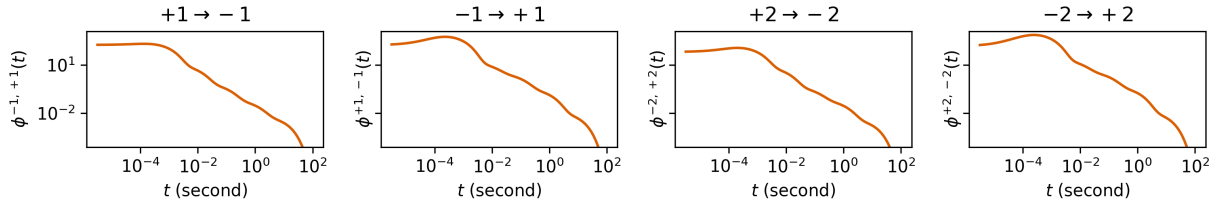


(a) BNP

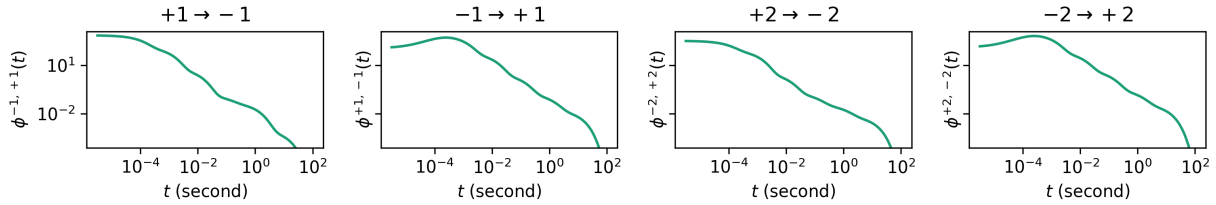


(b) NOKIA

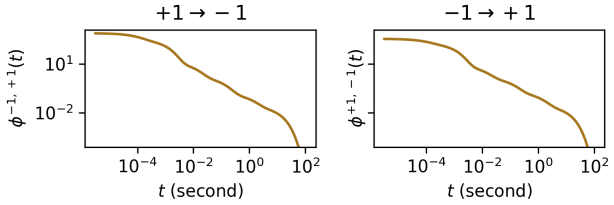
Figure 3.C.1 – **Relative kernel integrated quantities** $\tilde{I}_{s,e}(e')$ for (a)BNP and (b) NOKIA. Each heatmap matrix corresponds to a stock and a value of s, displaying $\tilde{I}_{s,e}(e')$ (defined by (3.4.3)) as a function of e (vertical axis) and e' (horizontal axis).



(a) BNP

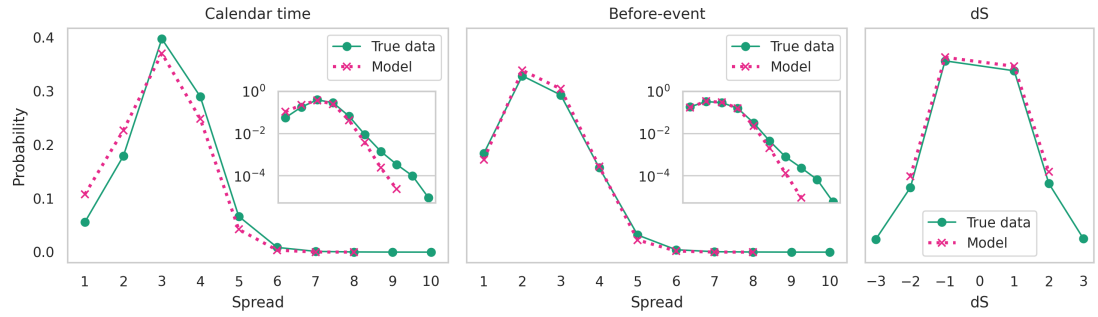


(b) NOKIA

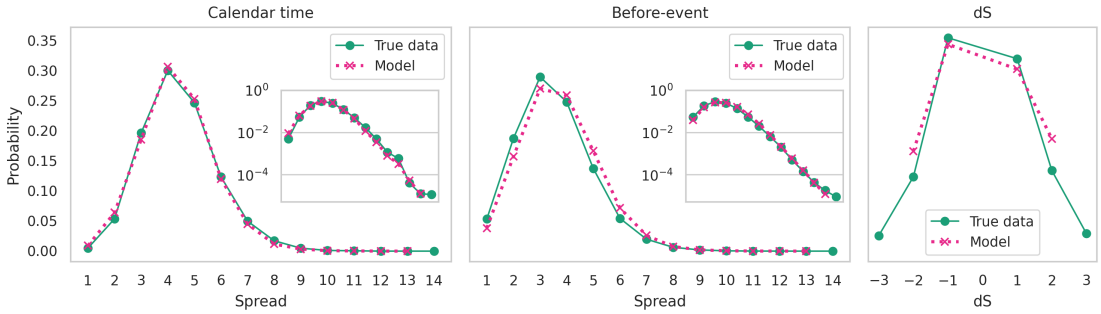


(c) CAC40 Future

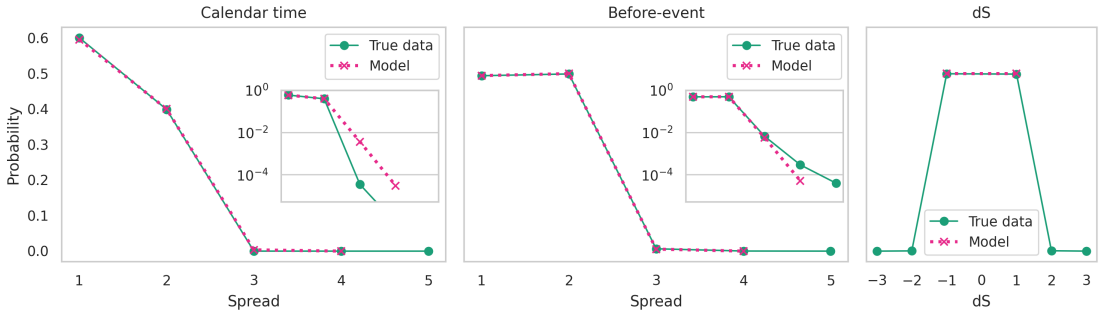
Figure 3.C.2 – **Hawkes kernel shapes** for (a) BNP, (b) NOKIA, and (c) CAC40 Future. In each figure we present four (two for CAC40 Future) contrariant Hawkes kernels (i.e., $\varphi^{e,e'}$ for $e = -e'$). Each kernel $\varphi^{e,e'}$ (labeled $e' \to e$ on the figure with $e, e' \in \mathcal{E}$) represents the influence of the past jumps of size e' on the occurrence probability of a future jumps of size e. Each kernel is represented as a sum of $L = 6$ exponentials, specifically $\varphi^{e,e'}(t)$, where $\varphi^{e,e'}(t) = \sum_{l=1}^{L} \alpha_l^{e,e'} \beta_l e^{-\beta_l t}$, where $\beta_l = \frac{1}{\tau_l}$ and τ_l takes values in the range $\{10^{-4}s, 10^{-3}s, \ldots, 10^{1}s\}$. All the kernels are displayed on a log-log scale and show a power-law behavior on a large range of scales (3, 4 or even 5 decades)



(a) BNP

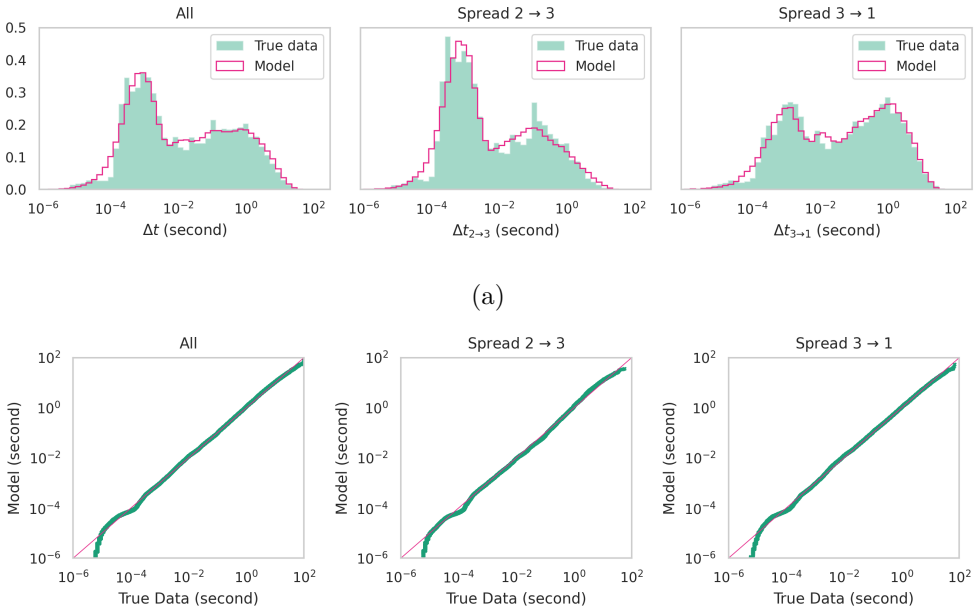


(b) NOKIA



(c) CAC40 Future

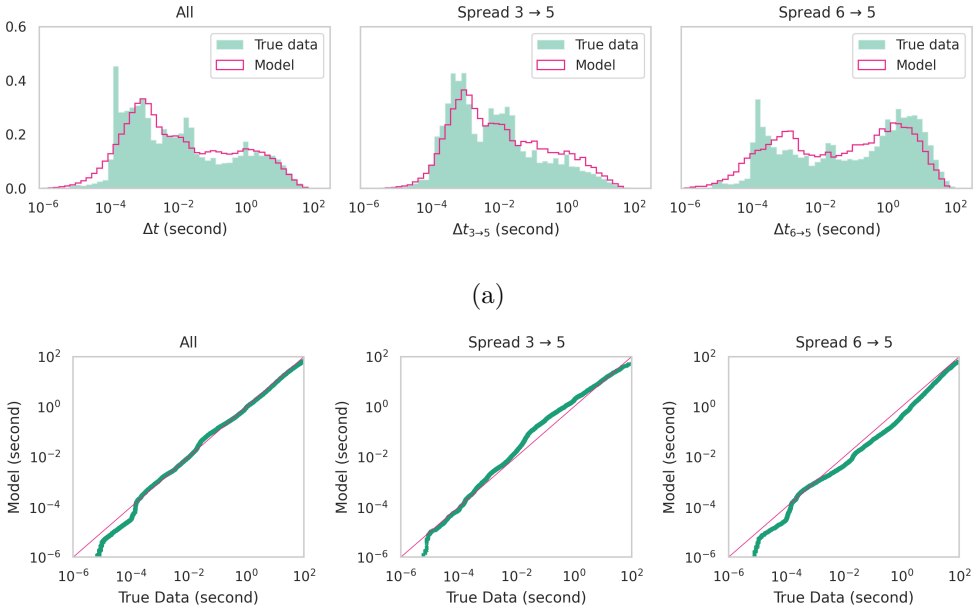
Figure 3.C.3 – **Spread distributions**, comparison between true data and the the data obtained through simulation of (a) BNP, (b) NOKIA, (c) CAC40 Future. The left-hand figures are the Calendar time distributions, the middle figures are the event time distributions, and the right-hand figures are the distributions of spread jumps size.



(a)

(b)

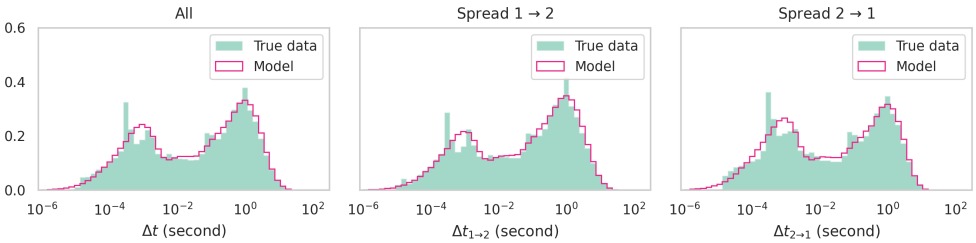
Figure 3.C.4 – **BNP Inter-event time distributions**. (a) Compare empirical inter-event-time distribution with simulated inter-event-time distribution. The x-axis is on log scale. (b) Log-log qq-plot of empirical inter-event-times (x-axis) *vs.* simulated inter-event-times by model (y-axis).



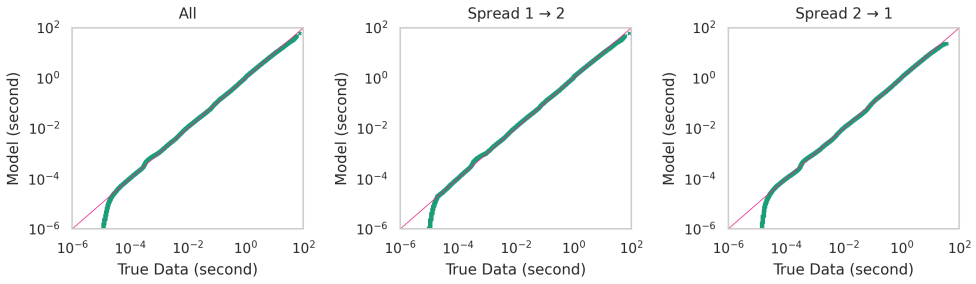
(a)

(b)

Figure 3.C.5 – **NOKIA Inter-event time distributions**. (a) Compare empirical inter-event-time distribution with simulated inter-event-time distribution. The x-axis is on log scale. (b) Log-log qq-plot of empirical inter-event-times (x-axis) *vs.* simulated inter-event-times by model (y-axis).

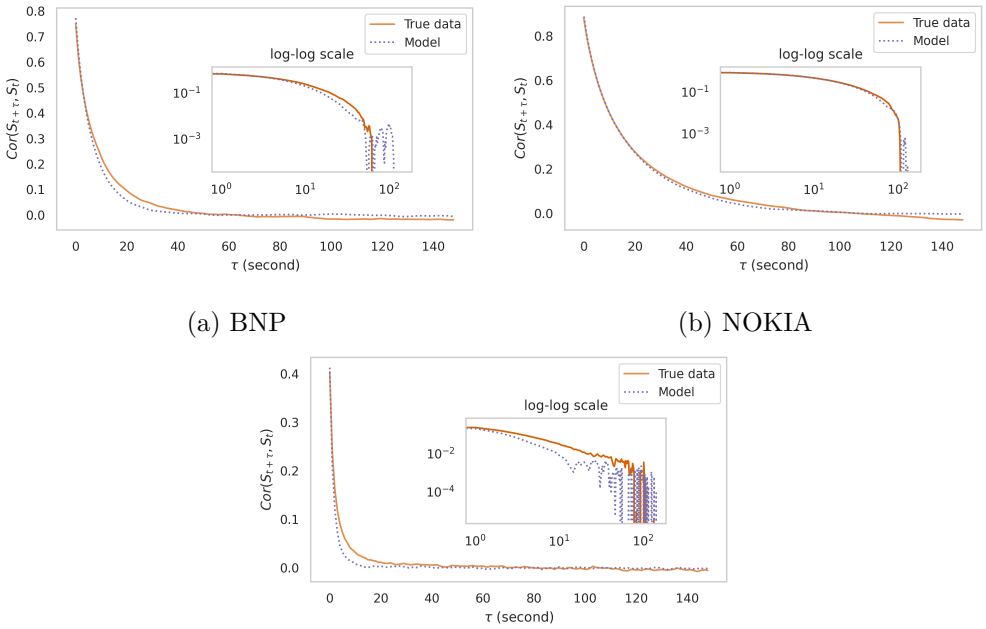


(a)



(b)

Figure 3.C.6 – **CAC40 index Future Inter-event time distributions** (a) Compare empirical inter-event-time distribution with simulated inter-event-time distribution. The x-axis is on log scale. (b) Log-log qq-plot of empirical inter-event-times (x-axis) *vs.* simulated inter-event-times by model (y-axis).



(a) BNP

(b) NOKIA

(c) CAC40 Future

Figure 3.C.7 – **Spread autocorrelation** for (a) BNP, (b) Nokia, (c) CAC40 index Future. True data versus model-simulated data. Each plot corresponds to a different asset. Quadratic variations were used for estimation. Eight 15min time-slots (between 10am and 12pm) were used everyday for true data in order to avoid intraday seasonal effects. 2 hours slots were used for model-simulated data.

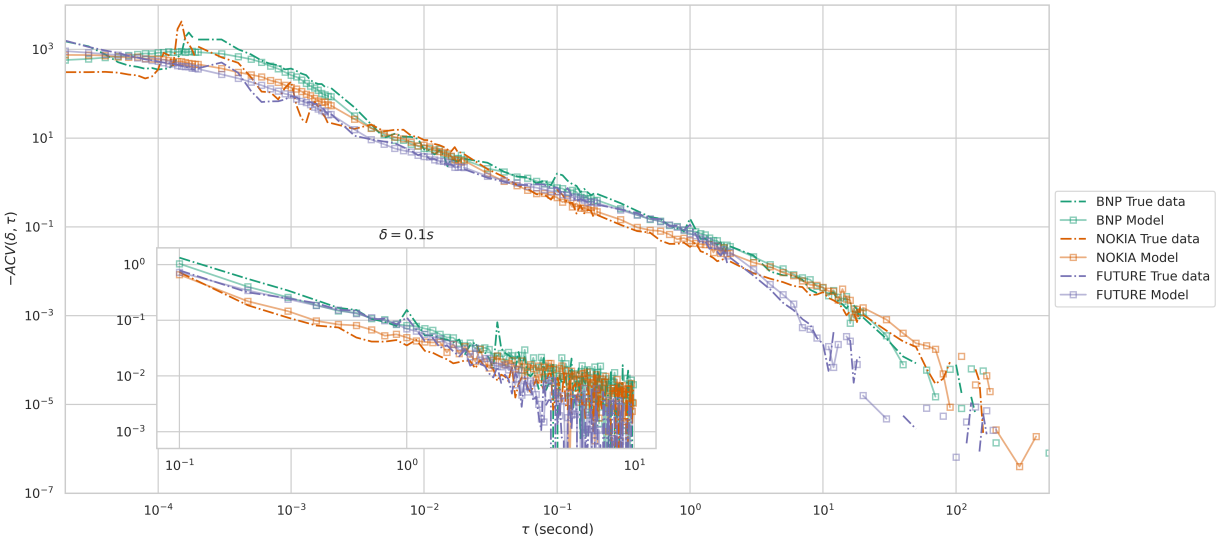


Figure 3.C.8 – **Autocovariance of spread increments**. The $-ACV(\delta, \tau)$ function for different assets. For each asset the curves that are displayed (in the main plot and in the inset) follow the same protocol as the one of Figure 3.9

CHAPTER 4

HAWKES PROCESS WITH SHOT NOISE MODEL

Joint work with E. Bacry, T. Deschatre, M. Hoffmann, J.-F. Muzy

We propose an expanded version of the Hawkes process model introduced by Bacry et al. (2013a), which we call the "Hawkes processes with shot noise" model. To capture the exogenous source of correlation, we introduce a latent dimension (shot noise) to the model. This latent dimension encodes the exogenous information that cannot be observed through the price processes, such as news and some specific agent behavior. We prove some limit theorems for this model and provide a non-parametric method for estimation. We also apply this model to real financial data.

Contents

4.1 - Introduction

Modeling the dynamics of asset prices has always been a crucial and also challenging task. At large scales, Brownian diffusion models are widely used. However, these continuous-time models have limitations when it comes to capturing the microstructural noise. On the other hand, Hawkes process, a finely grained model, has the ability to capture the main stylized facts of high-frequency data, such as Signature plots and the Epps Effect, as demonstrated in Bacry et al. (2013a). Hawkes process is a class of multivariate point processes, introduced by Hawkes (1971a,b) in the seventies. Since then, it has become very popular across various domains, including in finance. A significant body of literature is devoted to this topic, starting with the pioneering work of Bowsher (2007). Research on asset price modeling focuses on both single-asset prices models and multi-asset prices models. Single-asset works include Bacry and Muzy (2014); Da Fonseca and Zaatour (2015), which involve studying volatility and price autocorrelation. Bacry et al. (2013a); Da Fonseca and Zaatour (2017) are examples involving multi-asset models, which studied the correlation between assets prices. In this work, we follow the work of Bacry et al. (2013a,b) and focus on multi-asset models as well as the correlation between asset prices.

What are the factors contributing to the correlation between two assets? In this work, we propose that correlation arises from two different sources. The first source is the endogenous source, which comes from the internal feedback mechanism of the price processes themselves. This self-impact leads to correlations between assets. The second source is the exogenous source, driven by the external information, such as the news and agent behavior. An noteworthy recent work to mention in this context is Marcaccioli et al. (2022). In this paper, the authors showed that the abnormal price movements following news releases (exogenous) exhibit different dynamical features from those arising spontaneously (endogenous). Similar research can also be found in Bouchaud (2011); Sornette (2006); Sornette and Helmstetter (2003).

In the context of correlation between assets, to illustrate the exogenous source, let us consider the example of Bobl and Bund. These two assets are highly correlated, because they represent futures contracts on the same underlying asset with different maturities. As a result, global information of the underlying asset, such as news, can affect the prices of both assets. Moreover, traders who engage with both assets may employ hedge and arbitrage strategies and thus trade on both assets simultaneously. We can refer to these traders as *cross-traders*.

In this study, we present two models referred to as the "Hawkes process with shot noise" models, or abbreviated as "shot noise models". These models are designed to capture the different sources of correlation between two assets. In addition to the multivariate Hawkes process proposed in Bacry et al. (2013a), we suppose the existence of a latent (multivariate) Poisson process. The two models are respectively called the **latent-behavior model** and the **latent-information model**. The latent-behavior model assumes that the exogenous source of correlation comes from the latent behavior of above-mentioned cross-traders, while the latent-information model assumes that the exogenous source of correlation comes from the latent information, such as news.

Let us consider a simplified scenario where we have two counting processes, denoted as $\bar{N}_1$ and $\bar{N}_2$. These processes count respectively the numbers of trades on Asset 1 and Asset 2.

- In the **latent-behavior model** (two-dimensional asset process), each observable process comprises two terms. Specifically, $\bar{N}_1 = N_1 + N_4$ and $\bar{N}_2 = N_2 + N_5$. Let λ_i denote the conditional intensity of N_i, for $i \in \{1, 2, 3\}$. N_3 is a latent Poisson process, representing the cross-trader decision which is not directly observable. Whenever a shot noise event N_3

takes place, it generates two events in both asset processes N_4, N_5, with a random delay. If we denote the point process N_3 as $\{t_i\}_{i\in\mathbb{N}}$, then N_4 and N_5 are respectively $\{t_i + \Delta_{1,i}\}_{i\in\mathbb{N}}$ and $\{t_i + \Delta_{2,i}\}_{i\in\mathbb{N}}$. $\Delta_{k,i}$ are independent random variables. The latent-behavior model is mathematically defined as:

$$
\begin{aligned}
N_1 : \lambda_1(t) &= \mu_1 + \int_0^t \varphi_{11}(t-s)\, d(N_1(s) + N_4(s)) + \int_0^t \varphi_{12}(t-s)\, d(N_2(s) + N_5(s)) \\
N_2 : \lambda_2(t) &= \mu_2 + \int_0^t \varphi_{22}(t-s)\, d(N_2(s) + N_5(s)) + \int_0^t \varphi_{21}(t-s)\, d(N_1(s) + N_4(s)) \\
N_3 : \lambda_3(t) &= \mu_3
\end{aligned}
$$

N_1, N_2 are the events partially generated exogenously and independently with rates μ_1 and μ_2 and partially generated endogenously, by the self-exciting mechanism.

- In the **latent-information model**, the latent process N_3 represents the latent information such as news. The conditional intensity of the assets processes are defined as:

$$
\begin{aligned}
\bar{N}_1 : \lambda_1(t) &= \mu_1 + \int_0^t \varphi_{11}(t-s)\, d\bar{N}_1(s) + \int_0^t \varphi_{12}(t-s)\, d\bar{N}_2(s) + \int_0^t \varphi_{13}(t-s)\, dN_3(s) \\
\bar{N}_2 : \lambda_2(t) &= \mu_2 + \int_0^t \varphi_{21}(t-s)\, d\bar{N}_1(s) + \int_0^t \varphi_{22}(t-s)\, d\bar{N}_2(s) + \int_0^t \varphi_{23}(t-s)\, dN_3(s) \\
N_3 : \lambda_3(t) &= \mu_3
\end{aligned}
$$

  We assume the Poisson process N_3 is the latent information, such as the news. In this scenario, the shot noise events will only affect the intensities of both assets processes, without superimposing events on them. The observable processes are $\bar{N}_1$ and $\bar{N}_2$.

In this work, we mainly focus on the first shot noise model, i.e., the latent-behavior model. As to the latent-information model, we will briefly discuss it in Section 4.6. Following Bacry et al. (2013b), we will also provide some limit theorems for the Hawkes process with shot noise models.

Since the shot noise process is latent, estimation of the parameters of these shot noise models are challenging. For the classical Hawkes process, Achab et al. (2017) conducted a non-parametric cumulant method (NPHC) to estimate norms of Hawkes kernels. In this work, we will demonstrate that the NPHC estimator remains effective for the shot-noise models. Other existing methods typically rely on EM algorithm, such as Linderman et al. (2017); Mei et al. (2019); Shelton et al. (2018). In the appendices, we will also provide a Sequential Monte Carlo EM method for the parameters estimation for the shot noise model with latent agent behavior.

Outline The rest of this chapter is organized as follows. Section 4.2 introduces a two-dimensional latent-agent-behavior shot noise model. Prior to that, we provide some common notation and a review of the bivariate delayed Poisson process. In Section 4.3, we extend the limit theorems presented in Bacry et al. (2013b) to the latent-behavior model. Section 4.4 focuses on the review of the NPHC estimation method and its application to the latent-behavior model. In Section 4.5, we further extend the model to higher dimensionality and apply it to the real financial data. Section 4.6 is devoted to a brief study on the latent-information shot noise model. Finally, in Section 4.7, we conclude this work and discuss some future research directions.

4.2 - Hawkes process with shot noise model (latent-behavior)

4.2.1 Notation and definitions

- Let $\mathbb{I}_{n\times m}$ denote a matrix in $\mathbb{R}^{n\times m}$ filled with 1
- I_m denotes an identity matrix in $\mathbb{R}^{m\times m}$
- $\Phi(t)$ is a matrix of Hawkes kernel functions
  - $\Psi(t) = \sum_{n=1}^{\infty} \Phi^{\star n}(t)$ where $\Phi^{\star n}(t) = \underbrace{\Phi \star \Phi \star \cdots \star \Phi}_{n \text{ times } \Phi}$
  - Kernel norm matrix $G := \|\Phi\| = \int_0^\infty \Phi(t)dt$
  - $R(t) = I_m\delta(t) + \Psi(t)$ where m is the dimension of the process and $\delta(t)$ is the Dirac function
  - $\boldsymbol{R} = \int_0^\infty R(t)dt$
- The operator $\langle \cdot, \cdot \rangle$ is defined by $\langle f, g \rangle := \int_0^\infty f(t)g(t)^\top dt$ for two matrices of functions f and g with $f, g \in L^2(\mathbb{R}_+)$.

4.2.2 Bivariate Delayed Poisson process

Suppose we have an unobservable main Poisson process N_X of rate μ. A bivariate delayed Poisson process (N_1, N_2) is constructed from the main Poisson process N_X (see Lawrance and Lewis (1975); Lewis (1972)). An event at time t in N_X produces an event at time $t + \Delta^{(1)}$ in N_1 and an event at time $t + \Delta^{(2)}$ in N_2. $\Delta^{(1)}$ and $\Delta^{(2)}$ are two independent random variables.

The following proposition is from Example 7.3(a) in Daley et al. (2003).

Proposition 4.1. *Suppose that $\Delta^{(1)}$ and $\Delta^{(2)}$ are two independent random variables with exponential distributions of parameters a_1 and a_2 respectively. Suppose that at time t, given the internal history $\mathcal{F}_{t-}$, $N_X(t) = m$, $N_1 = n_1, N_2 = n_2$, where necessarily $m \geqslant \max\{n_1, n_2\}$. Then the intensity of (N_X, N_1, N_2) is given by*

$$\begin{aligned} \lambda_X(t) &= \mu \\ \lambda_1(t) &= a_1(m - n_1) \\ \lambda_2(t) &= a_2(m - n_2) \end{aligned} \tag{4.2.1}$$

The margin intensity of N_i is given by $\lambda_i(t) = \mu(1 - e^{-a_i t})$. When $t \to \infty$, the marginal processes become simple stationary Poisson processes of rate μ.

Remark 4.1. *We can also consider (N_X, N_1, N_2) in Proposition 4.1 as a multivariate Hawkes process. In this case, the conditional intensity can be written as*

$$\begin{aligned} \lambda_X(t) &= \mu \\ \lambda_i(t) &= \int_0^t a_i(\, dN_X(s) - \, dN_i(s)), \ \textit{for } i = 1, 2 \end{aligned} \tag{4.2.2}$$

Therefore the kernel matrix is

$$\Phi_0(t) \equiv \Phi_0 = \begin{pmatrix} 0 & 0 & 0 \\ a_1 & -a_1 & 0 \\ a_2 & 0 & -a_2 \end{pmatrix} \tag{4.2.3}$$

In the context of the Hawkes process framework, we define the function:

$$R_0(t) = \sum_{n=0}^{\infty} \Phi_0^{\star n}(t)$$

Here, $\Phi_0^{\star n}(t)$ represents the result of convolving the function $\Phi_0(t)$ with itself $n-1$ times using matrix convolutions. Let $A(t)$ and $B(t)$ denote two square matrices of functions, and $(A \star B)_{ij}(t)$ represents the matrix convolution operation between A and B. Specifically, $(A \star B)_{ij}(t)$ is given by: $(A \star B)_{ij}(t) = \sum_k \int_0^t A_{ik}(s) B_{kj}(t-s) ds$. Specifically, $A^{\star 0}(t) = \delta_0(t)\boldsymbol{I}$, where $\delta_0(t)$ is the Dirac delta function and $\boldsymbol{I}$ is the identity matrix.

By using an induction argument, we can demonstrate that $\Phi_0^{\star n}(t)$ can be expressed as follows:

$$\Phi_0^{\star n}(t) = \Phi_0^n \frac{t^{n-1}}{(n-1)!}$$

Here, Φ_0^n represents the matrix inner product performed n times. By calculation, we find that:

$$\Phi_0^n = (-1)^{n+1} \begin{pmatrix} 0 & 0 & 0 \\ (a_1)^n & -(a_1)^n & 0 \\ (a_2)^n & 0 & -(a_2)^n \end{pmatrix}$$

Consequently, we have, $\mathbb{R}_0(t) = \delta_0(t)\boldsymbol{I} + \Phi_0 e^{-\Phi_0 t}$ and

$$\boldsymbol{R}_0 = \int_0^{\infty} R_0(t) dt = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

This result is independent of the values of a_1 and a_2.

4.2.3 $(2 \times 2 + 1)$-dimensional Hawkes process with shot noise

Consider a simple scenario where we have two distinct assets, and we can observe the moments when their prices change. We assign the event type that moves the price of Asset 1 as $e = 1$, and the event type that changes the price of Asset 2 as $e = 2$. The point processes associated to these two event types are respectively denoted as $\bar{N}_1$ and $\bar{N}_2$.

In this work, we assume the existence of an unobservable exogenous Poisson process. We introduce a delay for each asset. When a noise event occurs at time t, this noise arrives at asset 1 with a

delay $\Delta^{(1)} \sim \text{Exp}(a_1)$ [1], i.e., appearing at stock 1 at time $t + \Delta_1$. Similarly, it arrives to stock 2 with a delay $\Delta^{(2)} \sim \text{Exp}(a_2)$, appearing at $t + \Delta^{(2)}$. Intuitively, this model can boost the macro correlation of the two prices thanks to these "common" shot noises. For example, when all the events originate from the shot noise, the macro correlation can even attain 1, which the classical model without shot noise cannot achieve. A 2-dimensional model is illustrated in Figure 4.2.1.

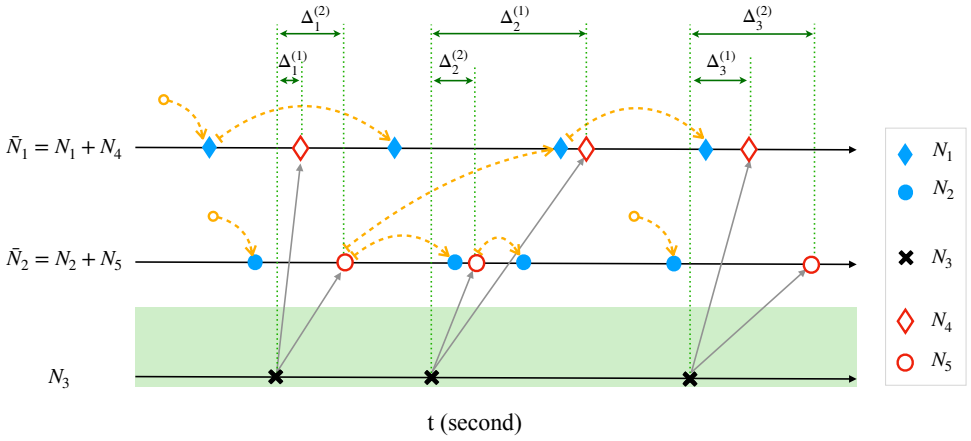


Figure 4.2.1 – An illustration of a 2D latent agent behavior shot noise model. For $i \in \{1, 2\}$, $\bar{N}_i = N_i + N_{i+3}$ represents the i-th price process which is observable, while N_3 stands for the latent shot noise process which is not directly observable. The yellow dashed arrows show the relation of generation. If an arrow points from a empty circle, it means that the event is an immigrant generated by an exogenous intensity for self-exciting processes. Otherwise, the arrow points to a child from its parent. The delay of the k-th shot noise on $\bar{N}_i$ is indicated by $\Delta_k^{(i)}$ (by our setting $\Delta_k^{(i)} \sim \text{Exp}(a_i)$). The common shot noise is represented by the greed shade.

Definition 4.1 ($(2 \times 2 + 1)$-dimensional Model)**.** *Suppose $a_1, a_2 > 0$, the intensity vector of $N = (N_i)_{i=1,2,..,5}$ is defined as*

$$\begin{cases} N_1 : \lambda_1(t) = \mu_1 + \int_0^t \varphi_{11}(t-s)\, d(N_1(s) + N_4(s)) + \int_0^t \varphi_{12}(t-s)\, d(N_2(s) + N_5(s)) \\ N_2 : \lambda_2(t) = \mu_2 + \int_0^t \varphi_{22}(t-s)\, d(N_2(s) + N_5(s)) + \int_0^t \varphi_{21}(t-s)\, d(N_1(s) + N_4(s)) \\ N_3 : \lambda_3(t) = \mu_3 \\ N_4 : \lambda_4(t) = a_1 \left(N_3(t) - N_4(t)\right) \\ N_5 : \lambda_5(t) = a_2 \left(N_3(t) - N_5(t)\right) \end{cases} \tag{4.2.4}$$

An oversimplified interpretation Let's consider a scenario where there are two assets and only three distinct types of agents operating in the market. Type-1 agents trade only the first asset and Type-2 agents trade only the second asset. However Type-3 agents possess both assets. Type-1 and Type-2 agents engage in trading activities using information from both assets, employing self- and cross-exciting kernels. On the other hand, Type-3 agents make their trading decisions based on a Poisson process (N_3), therefore independently to the events of both assets. When Type-3 agents decides to trade, they respond to both assets with independent random delays $\text{Exp}(a_i)$. However,

1. The probability density function for $\text{Exp}(a)$ is ae^{-at}

it's important to note that in the market, Type-1 and Type-2 agents are unable to differentiate between events generated by Types-1,2 or Type-3 agents. Consequently, they respond to both types of events using the same kernel (the reason for the term $\varphi_{ij}(t-s)\,d[N_j(s)+N_{j+3}(s)]$).

Let $\varphi_H = \begin{pmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{pmatrix}$, the kernel matrix for the complete model is $\Phi = \begin{pmatrix} \varphi_H & 0_{2\times 1}, \varphi_H \\ 0_{3\times 2} & \Phi_0 \end{pmatrix}$ where Φ_0 is defined in Equation (4.2.3). We denote μ_H to be $\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and $\psi_H(t) = \sum_{n=1}^{\infty} \varphi_H^{\star n}(t)$.

Proposition-definition 1. *Let* $\Gamma = \begin{pmatrix} -a_1 & 0 \\ 0 & -a_2 \end{pmatrix}$ *and* $\gamma = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$.

For $t \in \mathbb{R}^+$,

$$\begin{aligned} \Psi(t) &= \sum_{n=1}^{\infty} \Phi^{\star n}(t) \\ &= \begin{pmatrix} \psi_H(t) & \psi_H \star \bar{\gamma}(t) & \psi_H(t) + \psi_H \star \bar{\Gamma}(t) \\ 0_{1\times 2} & 0 & 0_{1\times 2} \\ 0_{2\times 2} & \bar{\gamma}(t) & \bar{\Gamma}(t) \end{pmatrix} \end{aligned} \tag{4.2.5}$$

where $\bar{\Gamma}(t) := \Gamma e^{\Gamma t} = \begin{pmatrix} -a_1 e^{-a_1 t} & 0 \\ 0 & -a_2 e^{-a_2 t} \end{pmatrix}$, $\bar{\gamma}(t) := e^{\Gamma t}\gamma = \begin{pmatrix} a_1 e^{-a_1 t} \\ a_2 e^{-a_2 t} \end{pmatrix}$

Therefore $R(t) = I_5\delta(t) + \Psi(t)$, *and*

$$\boldsymbol{R} = \int_0^{\infty} R(t)dt = \begin{pmatrix} \boldsymbol{R}_H & (\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 1} & 0_{2\times 2} \\ 0_{1\times 2} & 1 & 0_{1\times 2} \\ 0_{2\times 2} & \mathbb{I}_{2\times 1} & 0_{2\times 2} \end{pmatrix} \tag{4.2.6}$$

Proof. For $n \in \mathbb{N}^+$,

$$\Phi^{\star n} = \begin{pmatrix} \varphi_H^{\star n} & \sum_{k=1}^{n} \varphi_H^{\star k} \star \Gamma^{\star(n-k-1)} \star \gamma & \sum_{k=1}^{n} \varphi_H^{\star k} \star \Gamma^{\star(n-k)} \\ 0_{1\times 2} & 0 & 0_{1\times 2} \\ 0_{2\times 2} & \Gamma^{\star(n-1)} \star \gamma & \Gamma^{\star n} \end{pmatrix}$$

Since $\Gamma^{\star n}(t) = \Gamma^n \dfrac{t^{n-1}}{(n-1)!}$, we have directly $\sum_{n=1}^{\infty} \Gamma^{\star n}(t) = \Gamma e^{\Gamma t}$

$$\sum_{k=0}^{\infty} \Gamma^{\star k} \star \gamma(t) = (I_2\delta(t) + \Gamma e^{\Gamma t}) \star \gamma(t) = (\int_0^t (I_2\delta(s) + \Gamma e^{\Gamma s})ds) \cdot \gamma = e^{\Gamma t} \cdot \gamma$$

$$\sum_{n=1}^{\infty}\sum_{k=1}^{n}\varphi_H^{\star k}\star\Gamma^{\star(n-k-1)}\star\gamma=\psi_H\star(\sum_{k=0}^{\infty}\Gamma^{\star k})\star\gamma=\psi_H\star(e^{\Gamma t}\cdot\gamma)$$

□

Corollary 3. *The mean intensity of the complete model is*

$$\mathbf{\Lambda}=\boldsymbol{R}\mu=\begin{pmatrix}\boldsymbol{R}_H & (\boldsymbol{R}_H-I_2)\mathbb{I}_{2\times1} & 0_{2\times2}\\ 0_{1\times2} & 1 & 0_{1\times2}\\ 0_{2\times2} & \mathbb{I}_{2\times1} & 0_{2\times2}\end{pmatrix}\begin{pmatrix}\mu_H\\ \mu_3\\ 0_{2\times1}\end{pmatrix}=\begin{pmatrix}\boldsymbol{R}_H\mu_H+(\boldsymbol{R}_H-I_2)\mathbb{I}_{2\times1}\mu_3\\ \mu_3\\ \mu_3\\ \mu_3\end{pmatrix}$$

We denote by Λ_H the mean intensity vector of $\begin{pmatrix}N_1 & N_2\end{pmatrix}^\top$, that is $\Lambda_H=\boldsymbol{R}_H\mu_H+(\boldsymbol{R}_H-I_2)\mathbb{I}_{2\times1}\mu_3$. Then the observable mean intensity is

$$\bar{\Lambda}=\Lambda_H+\begin{pmatrix}\mu_3\\ \mu_3\end{pmatrix}=\boldsymbol{R}_H\begin{pmatrix}\mu_1+\mu_3\\ \mu_2+\mu_3\end{pmatrix}.$$

4.3 - Limit theorems

4.3.1 The law of large number and the central limit theorem

Consider the following assumption:

For all $i,j\in\{1,2\}$, $\|\varphi_{H,ij}\|=\int_0^\infty\varphi_{H,ij}(t)dt<\infty$ and the matrix $G_H=\|\varphi_H\|$ has spectral radius smaller than 1

(A4-1)

The observable process $\bar{N}=\begin{pmatrix}\bar{N}_1\\ \bar{N}_2\end{pmatrix}=\begin{pmatrix}N_1+N_4\\ N_2+N_5\end{pmatrix}$

Theorem 4.1. *Assume that (A4-1) holds. Then $N_t\in L^2(P)$ for $t\in\mathbb{R}^+$, and we have*

$$\sup_{v\in[0,1]}\|\frac{1}{T}N_{Tv}-v\boldsymbol{R}\mu\|\longrightarrow0$$

as $T\to\infty$ almost surely and in $L^2(P)$.

For observable process $\bar{N}$

$$\sup_{v\in[0,1]}\left\|\frac{1}{T}\bar{N}_{Tv}-v\bar{\Lambda}\right\|\longrightarrow0$$

where $\bar{\Lambda}$ is defined in Corollary 3.

Theorem 4.2. *Assume that (A4-1) holds. We have*

$$\frac{1}{\sqrt{T}}\left(N_{Tv} - \mathbb{E}[N_{Tv}]\right) \to \boldsymbol{R}\Sigma^{1/2}W_v, \text{ for } v \in [0,1]$$

in law for the Skorokhod topology as $T \to \infty$. $(W_v)_{v\in[0,1]}$ is a standard 5-dimensional Brownian motion. Σ is the diagonal matrix such that $\Sigma_{ii} = \Lambda_i$ where $\Lambda = R\mu$.

As the elements of the last two columns of R are all 0,

$$\frac{1}{\sqrt{T}}\left(N_{Tv} - \mathbb{E}[N_{Tv}]\right) \to \begin{pmatrix} \boldsymbol{R}_H \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} \begin{pmatrix} W_{1,v} \\ W_{2,v} \end{pmatrix} + (\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 1}\mu_3 W_{3,u} \\ \mu_3 W_{3,v} \\ \mu_3 W_{3,v} \\ \mu_3 W_{3,v} \end{pmatrix} \text{ for } v \in [0,1]$$

Therefore, for the observable process $\bar{N}$

$$\frac{1}{\sqrt{T}}\left(\bar{N}_{Tv} - \mathbb{E}[\bar{N}_{Tv}]\right) \to \boldsymbol{R}_H \begin{pmatrix} \Lambda_1 W_{1,v} + \mu_3 W_{3,v} \\ \Lambda_2 W_{2,v} + \mu_3 W_{3,v} \end{pmatrix} \text{ for } v \in [0,1]$$

Corollary 4. *Suppose that (A4-1) holds and furthermore we have the following restriction on φ_H,*

$$\boxed{\int_0^\infty \varphi_H(t)t^{1/2}dt < \infty} \tag{A4-2}$$

Then $\sqrt{T}(\frac{1}{T}N_{Tv} - v\boldsymbol{R}\mu) \to \boldsymbol{R}\Sigma^{1/2}W_v, v \in [0,1]$ in law for the Skorokhod topology when $T \to \infty$. Moreover,

$$\sqrt{T}(\frac{1}{T}\bar{N}_{Tv} - v\bar{\Lambda}) \to \boldsymbol{R}_H \begin{pmatrix} \Lambda_1 W_{1,v} + \mu_3 W_{3,v} \\ \Lambda_2 W_{2,v} + \mu_3 W_{3,v} \end{pmatrix}$$

4.3.2 Empirical covariation across time scales

For $N \in \mathbb{N}^5$, set $X_t = N_t - \mathbb{E}[N_t]$. On a time interval $[0,T]$, for $0 < \Delta < T$, the empirical covariance matrix of N is defined as

$$\mathbf{V}_{\Delta,T}(N) = \frac{1}{T}\sum_{i=1}^{\lfloor T/\Delta \rfloor} \left(X_{i\Delta} - X_{(i-1)\Delta}\right)\left(X_{i\Delta} - X_{(i-1)\Delta}\right)^\top \tag{4.3.1}$$

Similarly, for $\bar{N} = \begin{pmatrix} N_1 + N_4 \\ N_2 + N_5 \end{pmatrix}$, set $\bar{X}_t = \bar{N}_t - \mathbb{E}[\bar{N}_t]$, the empirical covariance matrix of $\bar{N}$ is

$$C_{\Delta,T}(\bar{N}) = \frac{1}{T}\sum_{i=1}^{\lfloor T/\Delta \rfloor} \left(\bar{X}_{i\Delta} - \bar{X}_{(i-1)\Delta}\right)\left(\bar{X}_{i\Delta} - \bar{X}_{(i-1)\Delta}\right)^\top$$

Theorem 4.3. *Let $(\Delta_T)_{T>0}$ be a family of positive real numbers. And suppose $\Delta_T/T \to 0$ as $T \to \infty$. We have*

$$\mathbf{V}_{\Delta_T,T}(N) - \mathbf{v}_{\Delta_T} \to 0 \text{ as } T \to \infty \text{ in } L^2$$

where

$$\mathbf{v}_\Delta = \int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ R(s)\Sigma R(t)^\top \, ds\, dt$$

In particular, for the observable process $\bar{N}$,

$$C_{\Delta_T,T}(\bar{N}) - c_{\Delta_T} \to 0 \text{ as } T \to \infty \text{ in } L^2$$

where

$$\begin{aligned} c_\Delta = &\int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ R_H(s)\bar{\Sigma}R_H(t)^\top \, ds\, dt \\ &+ \mu_3 \int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ \Big(R_H(s) \star \bar{\Gamma}(s)\Big) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \Big(R_H(t) \star \bar{\Gamma}(t)\Big)^\top ds\, dt \end{aligned} \tag{4.3.2}$$

Corollary 5 (Macroscopic covariance)**.**

$$\lim_{\Delta\to\infty} c_\Delta = \boldsymbol{R}_H \begin{pmatrix} \bar{\Lambda}_1 & \mu_3 \\ \mu_3 & \bar{\Lambda}_2 \end{pmatrix} \boldsymbol{R}_H^\top$$

where $\bar{\Lambda} = \begin{pmatrix} \bar{\Lambda}_1 & \bar{\Lambda}_2 \end{pmatrix}^\top$ is defined in Corollary 3.

Corollary 6 (Microscopic covariance)**.** *Assume that $a_1 = a_2 = a \gg 1$, for $\Delta \sim O(a)$*

$$\begin{aligned} c_\Delta &= \frac{1}{\Delta} Cov(\bar{N}_{t+\Delta} - \bar{N}_t, \bar{N}_{t+\Delta} - \bar{N}_t) \\ &= \begin{pmatrix} \bar{\Lambda}_1 & 0 \\ 0 & \bar{\Lambda}_2 \end{pmatrix} + \left(1 - \frac{1 - e^{-a\Delta}}{a\Delta}\right) \begin{pmatrix} 0 & \mu_3 \\ \mu_3 & 0 \end{pmatrix} + \Delta\tilde{c} + o(\Delta) \end{aligned}$$

where

$$\tilde{c} = \Big(\frac{1}{2}\psi_H(0) \begin{pmatrix} \bar{\Lambda}_1 & \mu_3 \\ \mu_3 & \bar{\Lambda}_2 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \bar{\Lambda}_1 & \mu_3 \\ \mu_3 & \bar{\Lambda}_2 \end{pmatrix} \psi_H(0)^\top + \int_0^\infty \psi_H(t) \begin{pmatrix} \bar{\Lambda}_1 & \mu_3 \\ \mu_3 & \bar{\Lambda}_2 \end{pmatrix} \psi_H(t)^\top dt\Big)$$

Remark 4.2. *If $c_\Delta^{12} = \frac{1}{\Delta} Cov(\bar{N}_{1,t+\Delta} - \bar{N}_{1,t}, \bar{N}_{2,t+\Delta} - \bar{N}_{2,t})$. For an given a,*

$$c_\Delta^{12} = \frac{1}{2}\mu_3 a\Delta + \tilde{c}^{12}\Delta + o(\Delta)$$

where

$$\begin{aligned} \tilde{c}^{12} = &\frac{1}{2}(\psi_{H,11}(0)\mu_3 + \psi_{H,12}(0)\bar{\Lambda}_2 + \psi_{H,21}(0)\bar{\Lambda}_1 + \psi_{H,22}(0)\mu_3) \\ &+ \bar{\Lambda}_1\langle\psi_{H,11}, \psi_{H,21}\rangle + \bar{\Lambda}_2\langle\psi_{H,12}, \psi_{H,22}\rangle + \mu_3\langle\psi_{H,11}, \psi_{H,22}\rangle + \mu_3\langle\psi_{H,12}, \psi_{H,21}\rangle \\ = &\mu_3 \left(\frac{1}{2}\psi_{H,11}(0) + \frac{1}{2}\psi_{H,22}(0) + \langle\psi_{H,12}, \psi_{H,21}\rangle + \langle\psi_{H,11}, \psi_{H,22}\rangle\right) \\ &+ \left(\frac{1}{2}\psi_{H,21}(0) + \langle\psi_{H,11}, \psi_{H,21}\rangle\right)\bar{\Lambda}_1 + \left(\frac{1}{2}\psi_{H,12}(0) + \langle\psi_{H,12}, \psi_{H,22}\rangle\right)\bar{\Lambda}_2 \end{aligned}$$

Then the shot noise term μ_3 is non-negligible and observable if

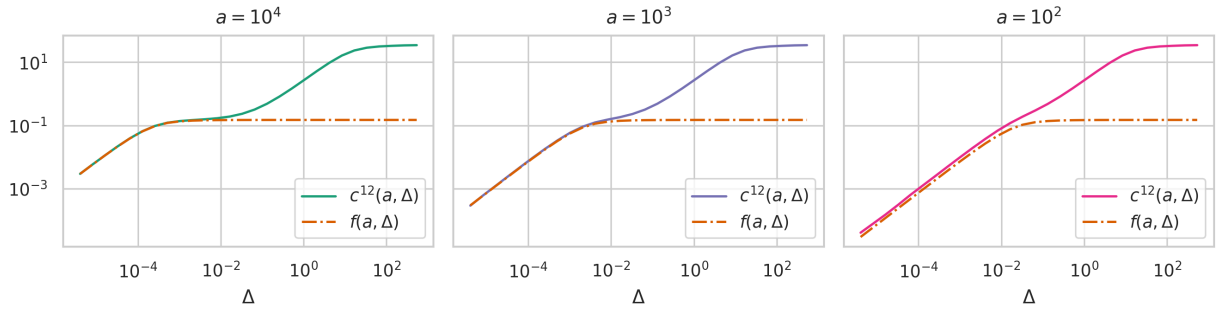
$$\mu_3 \left(\frac{a}{2} + \frac{1}{2}\psi_{H,11}(0) + \frac{1}{2}\psi_{H,22}(0) + \langle \psi_{H,12}, \psi_{H,21} \rangle + \langle \psi_{H,11}, \psi_{H,22} \rangle \right)$$
$$\gg \left(\frac{1}{2}\psi_{H,21}(0) + \langle \psi_{H,11}, \psi_{H,21} \rangle \right) \bar{\Lambda}_1 + \left(\frac{1}{2}\psi_{H,12}(0) + \langle \psi_{H,12}, \psi_{H,22} \rangle \right) \bar{\Lambda}_2$$

If we suppose furthermore that $\psi_H(0)$ are small compared to $\langle \psi_H, \psi_H \rangle$, then the following condition for a if sufficient to have an observable microscopic shot noise

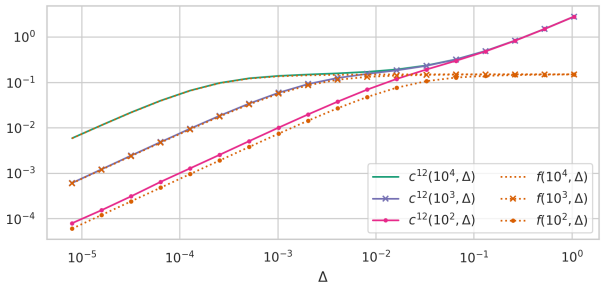
$$\frac{a}{2} \gg \frac{1}{\mu_3} \left(\langle \psi_{H,11}, \psi_{H,21} \rangle \bar{\Lambda}_1 + \langle \psi_{H,12}, \psi_{H,22} \rangle \bar{\Lambda}_2 \right) - \langle \psi_{H,12}, \psi_{H,21} \rangle - \langle \psi_{H,11}, \psi_{H,22} \rangle$$

Example 3. *Let us consider a model with the following parameters:*

- $\mu_H = \begin{pmatrix} 0.2 & 0.1 \end{pmatrix}$ *and* $\mu_3 = 0.15$
- $\varphi_{H,ij}(t) = \alpha_{ij} e^{-\beta t}$ *where* $\alpha = \begin{pmatrix} 0.8 & 0 \\ 0.3 & 0.6 \end{pmatrix}$ *and* $\beta = 1$
- $a_1 = a_2 = a$



(a) Comparison of c^{12} and f curves for three different values of a



(b) Comparison of three different values of a in one single plot

Figure 4.3.1 – Microscopic covariance for different values of a. The solid lines indicate the covariance curves $c^{12}(a, \Delta) = \frac{1}{\Delta} Cov(\bar{N}_{1,t+\Delta} - \bar{N}_{1,t}, \bar{N}_{2,t+\Delta} - \bar{N}_{2,t})$, while the dashed lines represent the functions $f(a, \Delta) = \mu_3(1 - \frac{1 - e^{-a\Delta}}{a\Delta})$. For each a, $c^{12}(a, \Delta)$ is empirically calculated from a simulated process of length 10^6 seconds, equivalent to approximately $3.68 \cdot 10^6$ events.

Through Figure 4.3.1, we can see that as a attains a significant magnitude, the platform on the quasi-constant segment of f becomes noticeable. Conversely, when a is small, the platform vanishes. Therefore, when assuming the value of a is large enough, this microscopic correlation curve can also be an heuristic criterion to detect the presence of the shot noise.

4.4 - Estimation: Apply NPHC on Hawkes with shot noise model

4.4.1 Review: Non-Parametric Hawkes Cumulant estimation methods (NPHC)

In the paper Achab et al. (2017), the authors developed an estimation technique by using the cumulants method. Given a classical d-variate Hawkes process N, let us define its conditional intensity vector as $\lambda_t = \mu + \int_0^t \varphi(t-s)\, dN_s$, where $\varphi(t)$ is the $d \times d$ kernel matrix. This paper provided an consistent estimator for the first, second and third-order integrated cumulants and then usd these cumulants to estimate the kernel matrix norm, i.e., $G = \int_0^\infty \Phi(t)dt$ (or equivalently $\boldsymbol{R}$ as $\boldsymbol{R} = (I - G)^{-1}$).

$$
\begin{aligned}
\Lambda_i &= \lim_{\delta \to 0} \frac{1}{\delta} \mathbb{E}[N_{i,t+\delta} - N_{i,t}] = \sum_{m=1}^{d} R_{im}\mu_m \\
C_{ij} &= \lim_{\delta \to 0} \frac{1}{\delta} \int_{\tau \in \mathbb{R}} Cov(N_{i,t+\delta} - N_{i,t}, N_{j,t+\delta+\tau} - N_{j,t+\tau}) \\
&= \sum_{m=1}^{d} \Lambda_m R_{im} R_{jm} \\
K_{ijk} &= \lim_{\delta \to 0} \frac{1}{\delta} \int_{\tau \in \mathbb{R}} \int_{\tau' \in \mathbb{R}} Skew(N_{i,t+\delta} - N_{i,t}, N_{j,t+\delta+\tau} - N_{j,t+\tau'}, N_{k,t+\delta+\tau'} - N_{k,t+\tau}) \\
&= \sum_{m=1}^{d} (R_{im}R_{jm}C_{km} + R_{im}C_{jm}R_{km} + C_{im}R_{jm}R_{km} - 2\Lambda_m R_{im}R_{jm}R_{km})
\end{aligned}
\tag{4.4.1}
$$

where $Skew$ stands for the function of coskewness i.e., $Skew(X, Y, Z) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])(Z - \mathbb{E}[Z])]$ for three random variables X, Y, Z.

Given a realization of a Hawkes process with asymptotically stationary increments N_t on $[0, T]$, let us note the realized process as $\{(t_k, e_k), k = 1, 2, \ldots, n\}$. The estimator of these three cumulants are given by

$$
\begin{aligned}
\hat{\Lambda}_i =& \frac{1}{T} \sum_{m=1}^{n} 1_{e_m = i} = \frac{N_{i,T}}{T} \\
\hat{C}_{ij} =& \frac{1}{T} \sum_{m=1}^{n} 1_{e_m = i} \left(N_{j,t_m+H} - N_{j,t_m-H} - 2H\hat{\Lambda}_j \right) \\
\hat{K}_{ijk} =& \frac{1}{T} \sum_{m=1}^{n} 1_{e_m = i} \left[\left(N_{j,t_m+H} - N_{j,t_m-H} - 2H\hat{\Lambda}_j \right) \left(N_{k,t_m+H} - N_{k,t_m-H} - 2H\hat{\Lambda}_k \right) \right] \\
& - \frac{\hat{\Lambda}_i}{T} \sum_{p,q=1}^{n} 1_{e_p = j, e_q = k} (2H - |t_p - t_q|)^+ + 4H^2 \hat{\Lambda}_i \hat{\Lambda}_j \hat{\Lambda}_k
\end{aligned}
\tag{4.4.2}
$$

for a H such that:

- each kernel $\varphi_{i,j}$ is essentially supported by $[0, H]$
- large enough s.t. the integration in the Eq 4.4.1 can pass to $[-H, H]$ with a small error
- small enough compared to T

The NPHC algorithm estimates the kernel norm matrix G by minimizing the "difference" between estimated cumulants and the theoretical cumulants.

4.4.2 Applying NPHC on Hawkes with shot noise model

The focus of this section is to provide the cumulant computations for the Hawkes model with shot noise and to demonstrate the validity of the cumulant method for our model. We note that the unknown variables in this model are R_H, μ_H and μ_3 which represent the integrated triggering kernel matrix, the exogenous intensity vector for the Hawkes process, and the rate for the shot noise Poisson process, respectively. It is worth noting that the parameters a_i are not included in the unknown variables, since they vanish in the macro cumulant terms.

In total, the number of parameters in the model is 7, where 4 is the number of parameters in R_H, 2 is the number of parameters in μ_H, and 1 for μ_3.

Mean intensity

The first-order cumulant of the full model is

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{R}_H & (\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 1} & 0_{2\times 2} \\ 0_{1\times 2} & 1 & 0_{1\times 2} \\ 0_{2\times 2} & \mathbb{I}_{2\times 1} & 0_{2\times 2} \end{pmatrix} \begin{pmatrix} \mu_H \\ \mu_3 \\ 0_{1\times 2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_H\mu_H + (\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 1}\mu_3 \\ \mu_3 \\ \mu_3\mathbb{I}_{2\times 1} \end{pmatrix}$$

Let us represent $\boldsymbol{R}_H\mu_H + (\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 1}\mu_3$ as Λ_H. The observable first-order cumulant is the sum of the mean intensity of the endogenous processes Λ_H and the shot noise processes $\mu_3\mathbb{I}_{2\times 1}$

$$\bar{\Lambda}(R_H, \mu_H, \mu_3) = \Lambda_H + \mu_3\mathbb{I}_{2\times 1} = \boldsymbol{R}_H \begin{pmatrix} \mu_1 + \mu_3 \\ \mu_2 + \mu_3 \end{pmatrix} \tag{4.4.3}$$

Covariance

The second-order cumulant of the full model is a 5×5 matrix

$$\boldsymbol{C} = \boldsymbol{R}_{5\times 5}\Sigma_{5\times 5}\boldsymbol{R}_{5\times 5}^{\intercal}$$

where $\Sigma_{5\times 5} = \begin{pmatrix} \Sigma_H & 0_{2\times 3} \\ 0_{3\times 2} & \mu_3 I_3 \end{pmatrix}$, Σ_H is the diagonal matrices with diagonal entries given by the vectors Λ_H, i.e., $\Sigma_{H,ii} = \Lambda_{H,i}$.

Further formula can be found by the following computations:

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{R}_H\Sigma_H\boldsymbol{R}_H^{\intercal} + \mu_3(\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 2}(\boldsymbol{R}_H^{\intercal} - I_2) & \mu_3(\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 1} & \mu_3(\boldsymbol{R}_H - I_2)\mathbb{I}_{2\times 2} \\ \mu_3\mathbb{I}_{1\times 2}(\boldsymbol{R}_H - I_2)^{\intercal} & \mu_3 & \mu_3\mathbb{I}_{1\times 2} \\ \mu_3\mathbb{I}_{2\times 2}(\boldsymbol{R}_H - I_2)^{\intercal} & \mu_3\mathbb{I}_{2\times 1} & \mu_3\mathbb{I}_{2\times 2} \end{pmatrix}$$

Then the second-order cumulant of the superposed processes (i.e. the observable covariance matrix) is

$$\begin{aligned} \bar{C}(R_H, \mu_H, \mu_3) = (C_{ij})_{i,j\in\{1,2\}} &= \Big(\sum_{k\in\{0,3\}} \sum_{k'\in\{0,3\}} \boldsymbol{C}_{i+k,j+k'} \Big)_{i,j\in\{1,2\}} \\ &= \boldsymbol{R}_H(\Sigma_H + \mu_3\mathbb{I}_{2\times 2})\boldsymbol{R}_H^{\intercal} \end{aligned} \tag{4.4.4}$$

Skewness

And if we denote the third-order cumulant of the 5-variate processes by $\boldsymbol{K}$ i.e.

$$\boldsymbol{K}_{ijk} = \sum_{m=1}^{5} (\boldsymbol{C}_{im}\boldsymbol{R}_{jm}\boldsymbol{R}_{km} + \boldsymbol{R}_{im}\boldsymbol{C}_{jm}\boldsymbol{R}_{km} + \boldsymbol{R}_{im}\boldsymbol{R}_{jm}\boldsymbol{C}_{km} - 2\boldsymbol{\Lambda}_m\boldsymbol{R}_{im}\boldsymbol{R}_{jm}\boldsymbol{R}_{km})$$

then for the superposed observable processes

$$\bar{K}(R_H, \mu_H, \mu_3) = (K_{ijk})_{i,j,k\in\{1,2\}} = (\sum_{d_1\in\{0,3\}} \sum_{d_2\in\{0,3\}} \sum_{d_3\in\{0,3\}} \boldsymbol{K}_{i+d_1,j+d_2,k+d_3})_{i,j,k\in\{1,2\}} \tag{4.4.5}$$

Remark 4.3 (consistency for $(2 \times 2 + 1) - D$)**.** *$\bar{\Lambda}$ leads to 2 equations, $\bar{C}$ results in 3 independent equations and $\bar{K}$ leads to 4 equations, therefore there are 9 independent equations, which are sufficient to uniquely determine the values of the variables R_H, μ_H and μ_3.*

In fact, suppose $\boldsymbol{R}_H = \begin{pmatrix} x & y \\ z & w \end{pmatrix}$

The observable cumulants are $\bar{\Lambda} = \begin{pmatrix} \bar{\Lambda}_1 \\ \bar{\Lambda}_2 \end{pmatrix} = \begin{pmatrix} (\mu_1+\mu_3)x + (\mu_2+\mu_3)y \\ (\mu_1+\mu_3)z + (\mu_2+\mu_3)w \end{pmatrix}$ *and*

$$\bar{C} = \begin{pmatrix} \bar{\Lambda}_1 x^2 + \bar{\Lambda}_2 y^2 + 2\mu_3 xy & \bar{\Lambda}_1 xz + \bar{\Lambda}_2 yw + \mu_3 xw + \mu_3 yz \\ \bar{\Lambda}_1 xz + \bar{\Lambda}_2 yw + \mu_3 xw + \mu_3 yz & \bar{\Lambda}_1 z^2 + \bar{\Lambda}_2 w^2 + 2\mu_3 zw \end{pmatrix}$$

$$\bar{K}_{111} = 3\bar{C}_{11}x^2 + 3\bar{C}_{12}y^2 - 2(\bar{\Lambda}_1 + \mu_3)x^3 - 2\bar{\Lambda}_2 y^3 - 2\mu_3(w+z)y^2 - 2\mu_3(x^2+2x-1)y + 2\mu_3 x$$

$$\bar{K}_{222} = 3\bar{C}_{22}w^2 + 3\bar{C}_{12}z^2 - 2\bar{\Lambda}_1 z^3 - 2(\bar{\Lambda}_2 + \mu_3)w^3 - 2\mu_3(x+y)z^2 - 2\mu_3(w^2+2w-1)z + 2\mu_3 w$$

The system of seven equations, namely $\bar{\Lambda}_1, \bar{\Lambda}_2, \bar{C}_{11}, \bar{C}_{12}, \bar{C}_{22}, \bar{K}_{111}, \bar{K}_{222}$, possesses already a unique solution. Consequently, according to Theorem 2.1 in Achab et al. (2017), we can conclude that NPHC for our shot noise model remains consistent.

The outline of the estimation algorithm can be found in Algorithm 1. We make a slight adjustment to the loss function to allow the error in the estimation of Λ.

Example 4. *Let us consider the following model with:*

- $\mu_H = \begin{pmatrix} 0.2 & 0.1 \end{pmatrix}$ *and* $\mu_3 = 0.15$
- $\varphi_{H,ij}(t) = \alpha_{ij}e^{-\beta t}$ *where* $\alpha = \begin{pmatrix} 0.8 & 0 \\ 0.3 & 0.6 \end{pmatrix}$ *and* $\beta = 100$
- $a_1 = a_2 = 1000$

Figure 4.4.1 illustrates the estimation results for this model. This figure empirically demonstrates the consistency of the NPHC method for our model.

Algorithm 1: Non Parametric Hawkes Cumulant estimation of G_H, μ_H and μ_3

Data:Observable processes (P_i) for $i \in \{1, 2\}$

Step 1: Estimate integrated cumulants from the observable data $\hat{\Lambda}, \hat{C}, \hat{K}$ (see (4.4.2))

Step 2: Design loss function

$$\mathcal{L}(\Theta) = \kappa_1 \|\Lambda(\Theta) - \hat{\Lambda}\| + \kappa_2 \|\bar{C}(\Theta) - \hat{C}\| + \kappa_3 \|\bar{K}^c(\Theta) - \hat{K}^c\|$$

$$\kappa_1 : \kappa_2 : \kappa_3 = (\|\hat{C}\| + \|\hat{K}^c\|) : (\|\hat{\Lambda}\| + \|\hat{K}^c\|) : (\|\hat{\Lambda}\| + \|\hat{C}\|)$$

Step 3: Estimate $\Theta = (R_H, \mu_H, \mu_3)$ by minimizing the loss function $\mathcal{L}(\Theta)$

$$\hat{\Theta} = (\hat{R}_H, \hat{\mu}_H, \hat{\mu}_3) = \arg\min_{\Theta} \mathcal{L}(\Theta)$$

Return: $\hat{G}_H = \mathbb{I}_2 - \hat{R}_H^{-1}$, $\hat{\mu}_H, \hat{\mu}_3$

4.5 - Extension to higher dimension and Application to finance

4.5.1 Extension

Let us extend our shot noise model to $(2 \times d + p)$ dimension, in this case, the event space

$$\mathcal{E} = \underbrace{\{N_{H,i}, i = 1, 2, \ldots, d\}}_{\mathcal{H}} \bigcup \underbrace{\{N_{X,k}, k = 1, 2, \ldots, p\}}_{\mathcal{X}} \bigcup \underbrace{\{N_{D,i}, i = 1, 2, \ldots, d\}}_{\mathcal{D}}$$

where N_H stands for the classical Hawkes process, N_X is the Poisson process noise and N_D is the delayed N_X. In particular, in $(2 \times 2 + 1)$-dimensional model, $N_H = (N_1, N_2)$, $N_X = N_3$ and $N_D = (N_4, N_5)$.

The conditional intensity function is then

$$\begin{cases} N_{H,i} : \lambda_{H,i}(t) = \mu_{H,i} + \sum_{j=1}^{d} \int_0^t \varphi_{ij}(t-s) d[N_{H,j}(s) + N_{D,j}(s)] \text{ for } i \in \mathcal{H} \\ N_{X,k} : \lambda_{X,k}(t) = \mu_{X,k} \text{ for } k \in \mathcal{X} \\ N_{D,i} : \lambda_{D,i}(t) = \sum_{k \in \mathcal{X}} a_{ik} [N_{X,k}(t) - N_{D,i}(t)] \text{ for } i \in \mathcal{D} \end{cases} \tag{4.5.1}$$

where a_{ik} is the parameter for the delay of the k-th shot noise on $N_{D,i}$.

In the case where $a_{ik} = \infty$, the k-th shot noise on $N_{D,i}$ occurs directly and without any delay. On the other hand, when $a_{ik} = 0$, the k-th shot noise on $N_{D,i}$ is completely absent.

The observable process in this general model is $\bar{N} = (\bar{N}_i)_{i \in \{1,2,\ldots,d\}}$ where

$$\bar{N}_i = N_{H,i} + N_{D,i}$$

Remark 4.4. *Under the general model (4.5.1), we can demonstrate the similar theorems as in Section 4. However, for the estimation, the consistency of NPHC is not always guaranteed. One must verify the number of independent equations in the system of cumulants and the number of unknown variables in each specific case.*

Example 5. *Let us consider the following model with $d = 4$ and $p = 2$:*

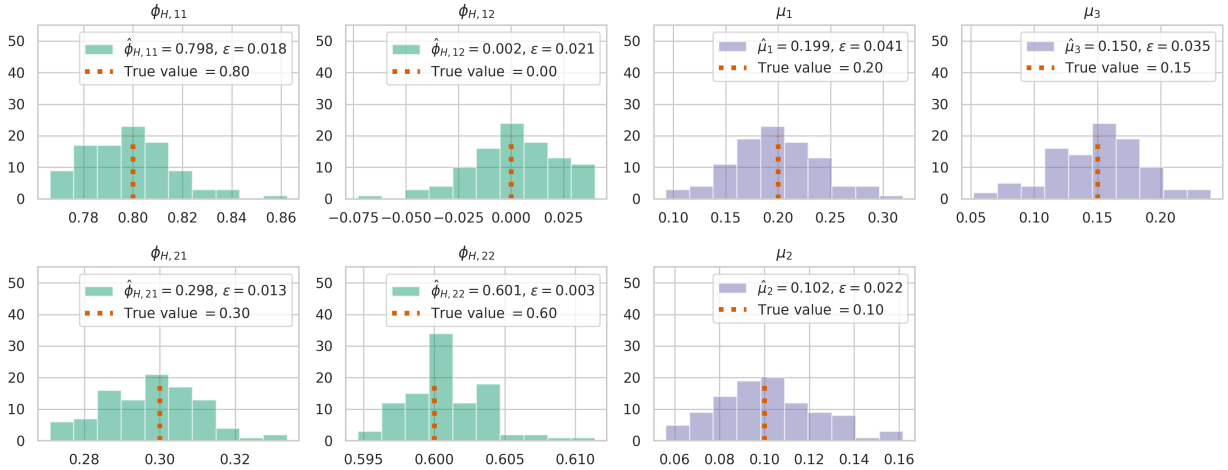


Figure 4.4.1 – Estimated kernel norms and baselines for Example 4. The red dashed vertical lines indicate the true values. The histograms represent the distributions of estimated values from 100 independent estimations. Each estimation is based on a simulated process spanning 10^6 seconds, equivalent to approximately $3.68 \cdot 10^6$ events. $\hat{\phi}_{H,ij}$ (resp. $\hat{\mu}$) indicates the estimated values for the kernel $\phi_{H,ij}$ (resp. μ). ϵ denotes the root-mean-square error between the estimated and true values. For example, in μ_3 figure, $\epsilon = \sqrt{\sum_{k=1}^{100}(\hat{\mu}_3^{(k)} - \mu_3)}$.

- $\mu_H = \begin{pmatrix} 0.1 & 0.1 & 0.15 & 0.15 \end{pmatrix}$ *and* $\mu_X = \begin{pmatrix} 0.2 & 0.2 \end{pmatrix}$
- $\varphi_{H,ij}(t) = \alpha_{ij} e^{-\beta t}$ *where* $\alpha = \begin{pmatrix} 0.01 & 0.15 & 0.4 & 0.1 \\ 0.15 & 0.01 & 0.1 & 0.4 \\ 0.2 & 0.05 & 0.01 & 0.4 \\ 0.05 & 0.2 & 0.4 & 0.01 \end{pmatrix}$ *and* $\beta = 100$
- $a_{11} = a_{31} = a_{22} = a_{42} = 1000$, *other* $a_{ik} = 0$

Figures 4.5.1, 4.5.2 and 4.5.3 show respectively the estimation results for the baselines μ, the matrix $\boldsymbol{R}_H$ and the kernel norms G_H. We can see that the estimations for μ and $\boldsymbol{R}_H$ exhibit a high degree of accuracy, while the estimation for G_H is biased. This bias originates from the computation of matrix inverses. In fact, the cumulant terms integrate the matrix $\boldsymbol{R}_H$ instead of G_H. Therefore, G_H can be viewed as a deduction from $\boldsymbol{R}_H$. When $\boldsymbol{R}_H$ has a spectral radius close to 1, it is very likely to have a large estimation error for G_H.

4.5.2 Bivariate assets price

Now let us proceed to a bivariate price model $P = (P_1, P_2)$ obtained from a 4-dimensional process

$$(P_1, P_2) = (\bar{N}_1 - \bar{N}_2, \bar{N}_3 - \bar{N}_4)$$

with $\bar{N}_{1,t}$ (resp. $\bar{N}_{2,t}$) represents the number of upward (resp. downward) price jumps at time t for Asset 1 and $\bar{N}_{3,t}$ (resp. $\bar{N}_{4,t}$) represents the number of upward (resp. downward) price jumps at time t for Asset 2. $\bar{N}_i = N_{H,i} + N_{D,i}$.

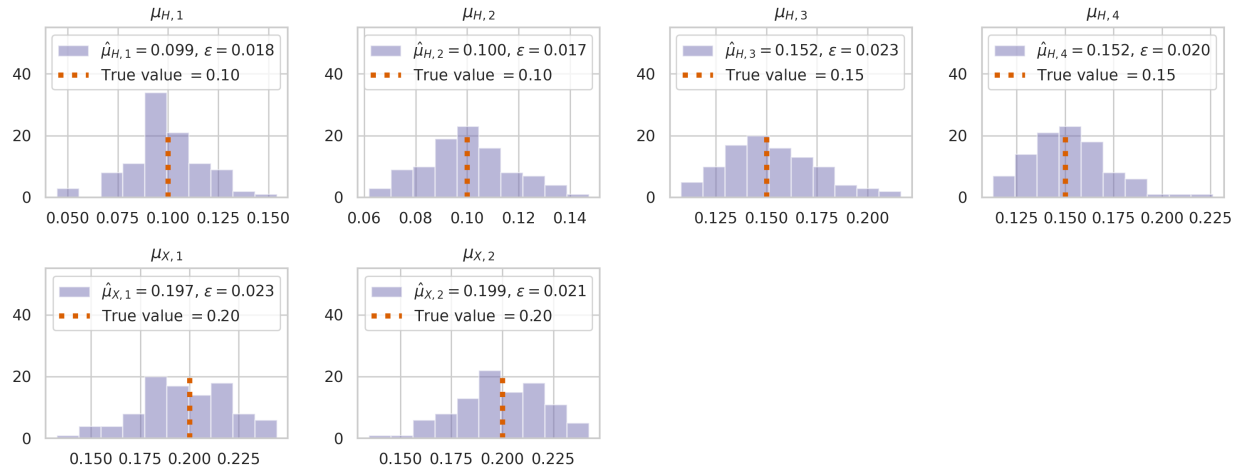


Figure 4.5.1 – Estimated baselines for Example 5. The red dashed vertical lines indicate the true values. The histograms represent the distributions of estimated values from 100 independent estimations. Each estimation is based on a simulated process spanning 10^6 seconds, equivalent to approximately $4 \cdot 10^6$ events.

Suppose that the 10 dimensional process N is defined as

$$N = (N_{H,1}, N_{H,2}, N_{H,3}, N_{H,4}, N_{X,1}, N_{X,2}, N_{D,1}, N_{D,2}, N_{D,3}, N_{D,4})$$

And its associated conditional intensities are as follows:

$$\begin{cases} N_{H,i} : \lambda_{H,i} = \mu_{H,i} + \sum_{j=1}^{4} \int_0^t \varphi_{ij}(t-s) d[N_{H,j}(s) + N_{D,j}(s)] \text{ for } i \in \{1,2,3,4\} \\ N_{X,k} : \lambda_{X,k}(t) = \mu_{X,k} \text{ for } k \in \{1,2\} \\ N_{D,i} : \lambda_{D,i}(t) = a_1[N_{X,1}(t) - N_{D,i}(t)] \text{ for } i \in \{1,3\} \\ N_{D,i} : \lambda_{D,i}(t) = a_2[N_{X,2}(t) - N_{D,i}(t)] \text{ for } i \in \{2,4\} \end{cases} \tag{4.5.2}$$

In other works, this is an extended shot noise model with 2 independent sources of Poisson processes noises $N_{X,1}, N_{X,2}$. $N_{X,1}$ produce two delayed processes on N_1 and N_3, i.e., the upward price jumps processes for the two assets, and $N_{X,2}$ produce two delayed processes on N_2 and N_4, i.e., the downward price jumps processes for the two assets.

Disentangling endogenous & exogenous The macroscopic covariance matrix can be split into two parts:

$$\begin{aligned} \bar{C} &= \boldsymbol{R}_H \Big[\Sigma_H + \mu_X \begin{pmatrix} I_2 & I_2 \\ I_2 & I_2 \end{pmatrix} \Big] \boldsymbol{R}_H^\top \\ &= \underbrace{\boldsymbol{R}_H diag(\boldsymbol{R}_H \mu_H) \boldsymbol{R}_H^\top}_{=:I_1} + \underbrace{\mu_X \boldsymbol{R}_H \Big[diag(\boldsymbol{R}_H \mathbb{I}_{4\times 1}) + \begin{pmatrix} 0 & I_2 \\ I_2 & 0 \end{pmatrix} \Big] \boldsymbol{R}_H^\top}_{=:I_2} \end{aligned} \tag{4.5.3}$$

where $diag(v)$ is the diagonal matrix with diagonal entries given by the vector v.

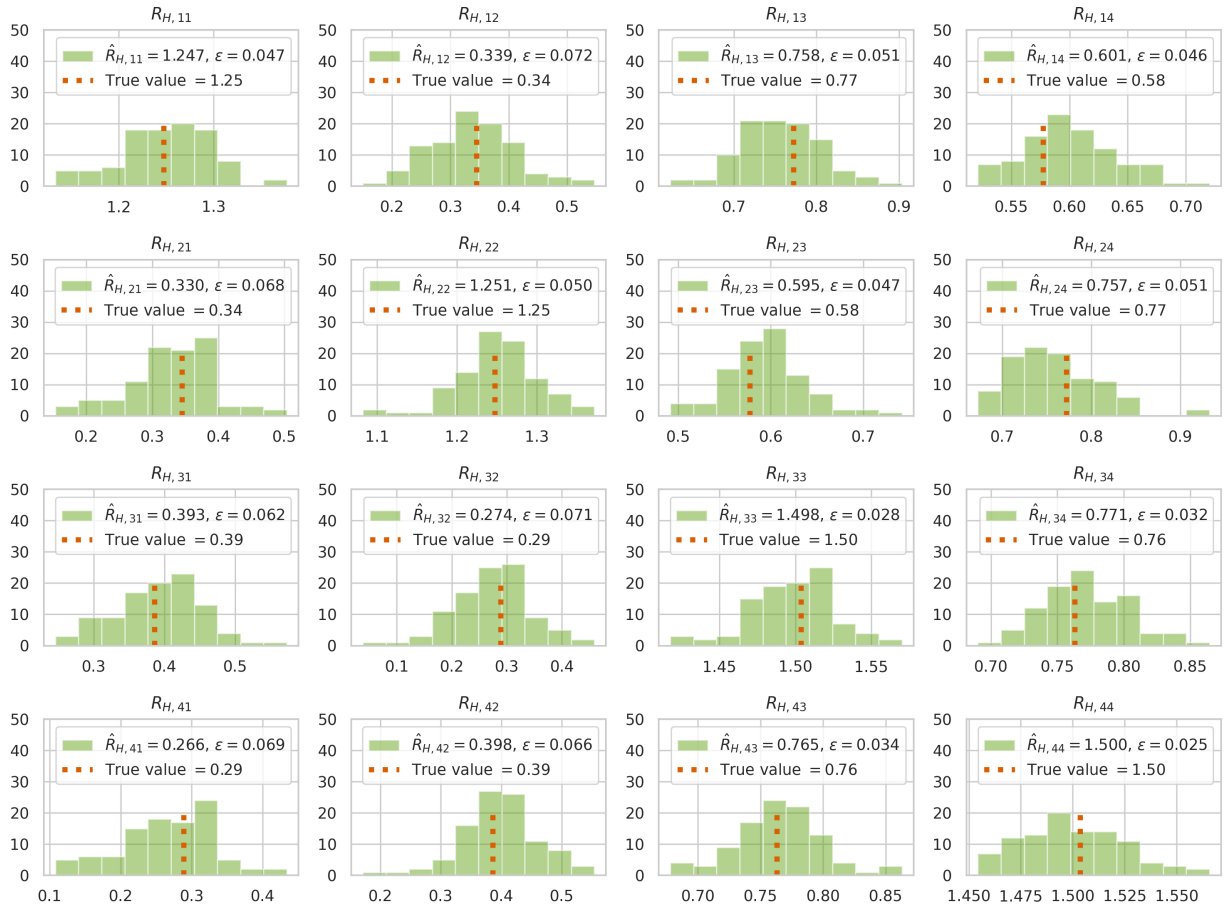


Figure 4.5.2 – Estimated R_H for Example 5. The red dashed vertical lines indicate the true values. The histograms represent the distributions of estimated values from 100 independent estimations. Each estimation is based on a simulated process spanning 10^6 seconds, equivalent to approximately $4 \cdot 10^6$ events.

Under the decomposition (4.5.3), I_1 is considered to be the endogenous part of the covariance matrix, as it is independent of the exogenous intensity μ_X. I_2 is the exogenous part.

Example 6 (BNP Paribas & Société Générale)**.** *To calibrate the latent-behavior model, we use high-frequency data from BNP Paribas and Société Générale, obtained from the CAC40 French Euronext Market. The dataset covers a period of 272 days starting from February 2017. We extract the price movements of both assets each day between 9 am and 11 am. To reduce the impact of varying mean intensities, we divide this time interval into ten 10-minute time slots. In total, we have approximately* $4.6 \cdot 10^6$ *events available for estimation.*

We set $H = 1s$*, and the estimated parameters are presented in Figure 4.5.4.*

The total correlation corresponding to $\bar{C}$ *in (4.5.3) is 16.1%, while the endogenous correlation corresponding to* I_1 *is 13.1%. This means that the contribution of exogenous factors to the correlation is 3%.*

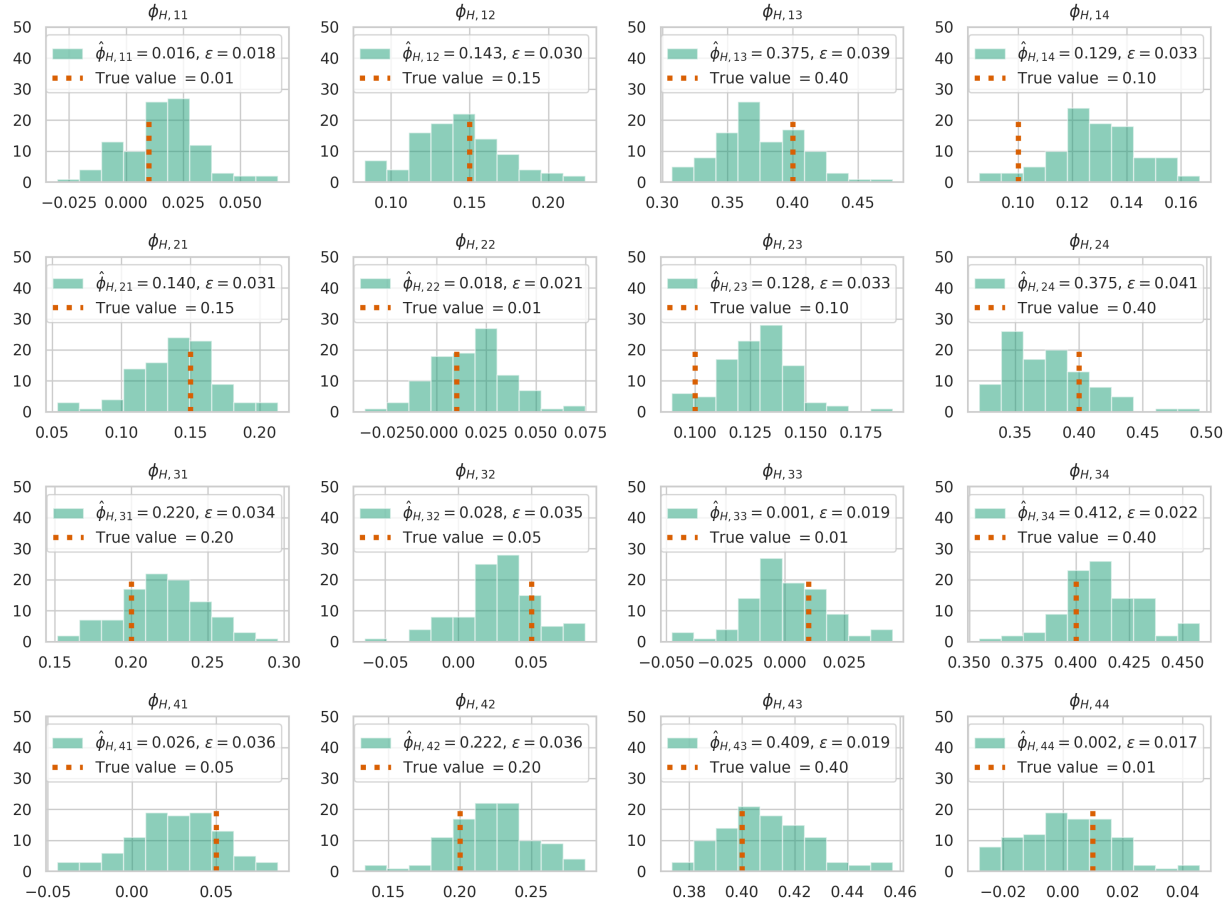


Figure 4.5.3 – Estimated $\|\Phi_H\|$ (or equivalently G_H) for Example 5. The red dashed vertical lines indicate the true values. The histograms represent the distributions of estimated values from 100 independent estimations. Each estimation is based on a simulated process spanning 10^6 seconds, equivalent to approximately $4 \cdot 10^6$ events.

4.6 - A variant of Hawkes process with shot noise model (latent information)

In this section, let us proceed to study the latent-information shot noise model. Let us start with a simple scenario where we have 2 different assets and we can track the times at which their prices change. We label the event type that moves the price of asset 1 as N_1, and the event type that changes the price of asset 2 as N_2. Additionally, we assume the existence of unobservable exogenous information, such as the news, that can affect the prices of both assets. We denote by N_3 this event type and consider it to follow a Poisson process. In this case the event space $\mathcal{E} = \{N_1, N_2, N_3\}$.

4.6.1 Model

In this model, N_1 and N_2 are two **observable** point processes, N_3 is a **latent** Poisson process which can impact the intensities of N_1 and N_2. The shot noise model (illustrated in Figure 4.6.1)

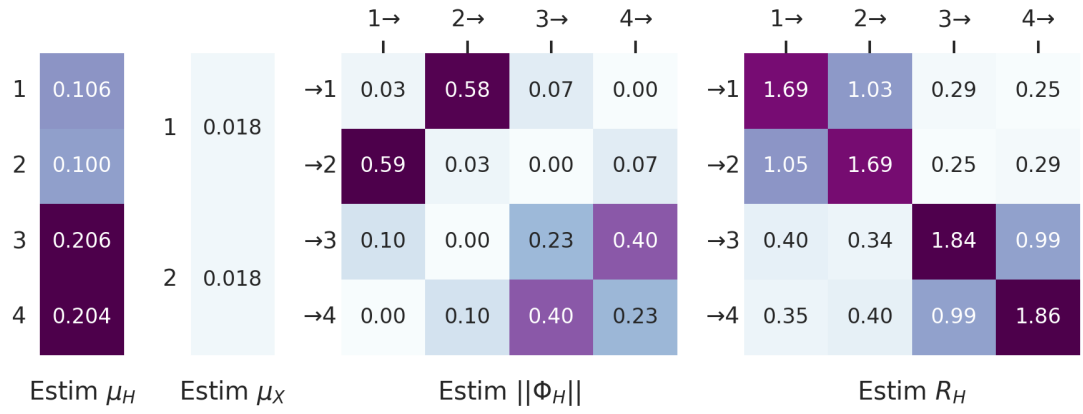


Figure 4.5.4 – Estimation of Hawkes kernel norms and baselines for BNP Paribas and Société Générale. "1" and "2" stand for the upward and downward price jumps for BNP Paribas while "3" and "4" stand for the upward and downward price jumps for Société Générale.

is defined as follows:

$$\begin{aligned} N_1 : \lambda_1(t) &= \mu_1 + \sum_{j\in\mathcal{E}} \int_0^t \varphi_{1j}(t-s)\, dN_j(s) \\ N_2 : \lambda_2(t) &= \mu_2 + \sum_{j\in\mathcal{E}} \int_0^t \varphi_{2j}(t-s)\, dN_j(s) \\ N_3 : \lambda_3(t) &= \mu_3 \end{aligned} \tag{4.6.1}$$

where

- μ_1 and μ_2 are the constant baseline intensities of N_1 and N_2 respectively, and μ_3 is the intensity of Poisson process N_3
- $\varphi_{ij}(t)$ is the impact kernel function of event type j on event type i, for $i \in \{1,2\}$ and $j \in \mathcal{E}$

Remark 4.5. *As a matter of fact, this model can also be interpreted as a Hawkes process with stochastic baselines.*

$$\lambda_i(t) = \tilde{\mu}_i(t) + \sum_{j\in\{1,2\}} \int_0^t \varphi_{ij}(t-s)\, dN_j(s)$$

where $\tilde{\mu}_i(t) = \mu_i + \displaystyle\int_0^t \varphi_{i,3}(t-s)\, dN_3(s)$, *for* $i \in \{1,2\}$.

$(d+p)$-dimensional process

Event space $\mathcal{E} = \underbrace{\{N_{H,i}, i = 1,2,\dots,d\}}_{\mathcal{H}} \cup \underbrace{\{N_{X,k}, k = 1,2,\dots,p\}}_{\mathcal{X}}$

$$\begin{aligned} &N_{H,i} : \lambda_{H,i}(t) = \mu_{H,i} + \sum_{j=1}^{d} \int_0^t \varphi_{H,ij}(t-s)\, dN_{H,j}(s) + \sum_{k=1}^{p} \int_0^t \varphi_{X,ik}(t-s)\, dN_{X,k}(s) \text{ for } i \in \{1,2,\dots,d\} \\ &N_{X,k} : \lambda_{X,k}(t) = \mu_{X,k} \text{ for } k \in \mathcal{X} \end{aligned} \tag{4.6.2}$$

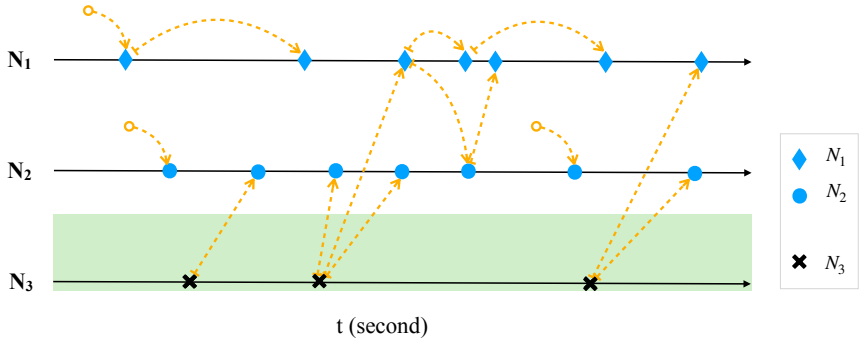


Figure 4.6.1 – An illustration of a (2+1)-dimensional latent information shot noise model. For $i = 1, 2$, N_i represents the i-th price process which is observable, while N_3 stands for the latent shot noise process which is not directly observable. The yellow dashed arrows show the relation of generation. If an arrow points from a empty circle, it means that the event is an immigrant generated by an exogenous intensity for self-exciting processes. Otherwise, the arrow points to a child from its parent. The shot noise process is represented by the greed shade.

Kernel function matrix

$$\Phi = \begin{pmatrix} \varphi_H & \varphi_X \\ 0_{p\times d} & 0_{p\times p} \end{pmatrix}$$

We denote $\|\Phi\|$ the matrix obtained by taking the integrals of the components of kernel matrix Φ, then

$$R = (\boldsymbol{I}_{d+p} - \|\Phi\|)^{-1} = \begin{pmatrix} (\boldsymbol{I}_d - \|\boldsymbol{\varphi}_H\|)^{-1} & (\boldsymbol{I}_d - \|\boldsymbol{\varphi}_H\|)^{-1}\|\boldsymbol{\varphi}_X\| \\ \boldsymbol{0}_{p\times d} & \boldsymbol{I}_p \end{pmatrix} =: \begin{pmatrix} R_H & R_X \\ \boldsymbol{0}_{p\times d} & \boldsymbol{I}_p \end{pmatrix}$$

4.6.2 NPHC Estimation

Let us denote the observable cumulant functions as $\bar{\Lambda}$, $\bar{C}$ and $\bar{K}$. The number of independent equations provided by these three cumulants is given by $d + \dfrac{d(d+1)}{2} + \dfrac{d^3 + 3d^2 + 2d}{6}$.

The parameters for estimation include $\|\varphi_H\|$, $\|\varphi_X\|$ and μ giving a total of $(d+p)(d+1)$ parameters. Among them, d^2 parameters come from $\|\varphi_H\|$, dk from $\|\varphi_X\|$, and $d+p$ from μ. Therefore, for a consistent estimation, one necessary condition is that the number of equations should be greater than the number of parameters. In other words: $p \leqslant \dfrac{d^3 + 5d}{6(d+1)}$. For example, when $d = 2$, we have $p \leqslant 1$, and when $d = 4$, we have $p \leqslant 2$.

Cumulant formula The theoretical cumulants are the functions on $\Theta = (R_S, R_X, \mu)$. In the complete model, expressing the first-order cumulants is straightforward and can be done as follows: First order cumulant (mean intensity)

$$\boldsymbol{\Lambda}(\Theta) = \begin{pmatrix} R_H\mu_H + R_X\mu_X \\ \mu_X \end{pmatrix} := \begin{pmatrix} \Lambda_H \\ \mu_X \end{pmatrix}$$

Second order cumulant (covariance)

$$\boldsymbol{C}(\Theta) = \begin{pmatrix} R_H \Sigma_H R_H^T + R_X \Sigma_X R_X^T & R_X \Sigma_X \\ \Sigma_X R_X^T & \Sigma_X \end{pmatrix} =: \begin{pmatrix} C_H(\Theta) & C_X(\Theta) \\ C_X(\Theta) & \Sigma_X \end{pmatrix}$$

where Σ_H and Σ_X are the diagonal matrices with diagonal entries given by the vectors Λ_H and Λ_X respectively.

Example 7. *Figure 4.6.2 illustrates the estimation results for the model with the following parameters:*

- $\mu = \begin{pmatrix} 0.5 & 0.4 & 0.1 \end{pmatrix}$
- $\varphi_{ij}(t) = \alpha_{ij} e^{-\beta t}$ *for* $i \in \{1,2\}, j \in \{1,2,3\}$ *where* $\alpha = \begin{pmatrix} 0.5 & 0.1 & 1 \\ 0.1 & 0.5 & 1 \end{pmatrix}$

The estimation results are less accurate than those in Examples 4and 5, especially for the kernels ϕ_{13} and ϕ_{23}. As this model has more parameters than the previous ones, there can be some mutual compensation between ϕ_{i3} and μ_3 within the context of statistical errors associated with the cumulants.

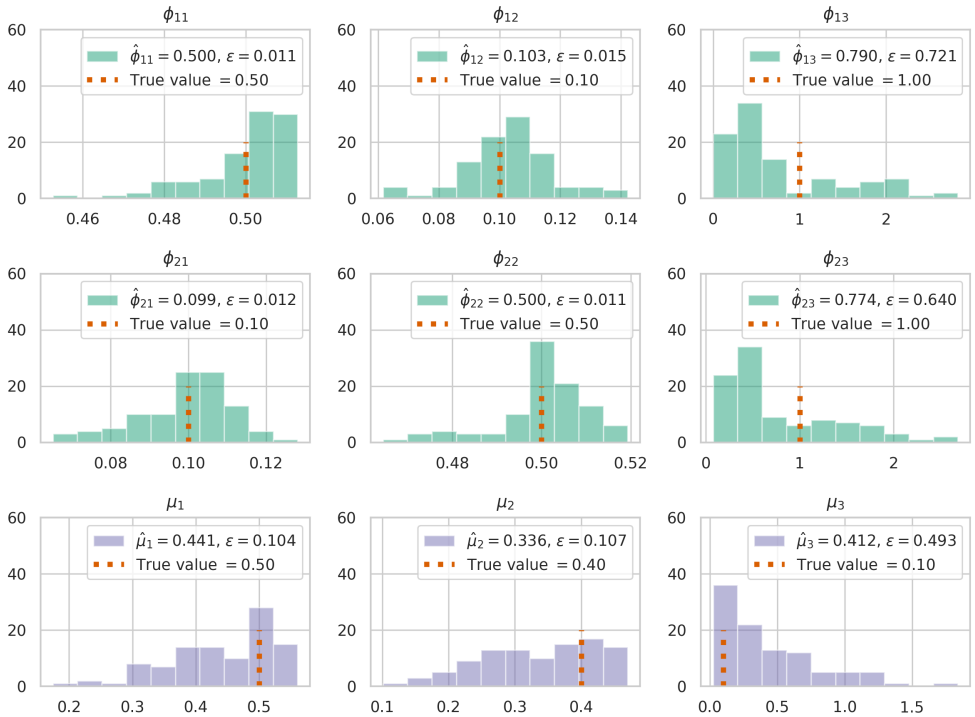


Figure 4.6.2 – Estimated kernel norms and baselines for Example 7. The red dashed vertical lines indicate the true values. The histograms represent the distributions of estimated values from 100 independent estimations. Each estimation is based on a simulated process spanning 10^6 seconds, equivalent to approximately $2.75 \cdot 10^6$ events.

4.7 - Conclusion and future research

This chapter introduced our ongoing work concerning the Hawkes process with shot noise models, with a specific focus on the first model, known as the latent-behavior model. Theoretically, we provided the limit theorems for this model. In the empirical part, we demonstrated the consistency of the NPHC method and illustrated the estimation accuracy through two examples. We also applied this model to the price processes of BNP Paribas and Société Générale and found a non-zero shot noise contribution. At the end of this chapter, we briefly introduced a variant of this model, called the latent-information model, for which the NPHC estimation method remains applicable.

In the future, we will delve deeper into the practical applications of these models to real data. We also plan to study EM estimation methods for these models, which can give us more information on the latent shot noise process (e.g., kernels' shapes, shot noise delays). Moreover, another prospect is constructing another model which combines both the latent-behavior model and the latent-information model.

APPENDIX

4.A - Proofs

The proofs of Theorem 4.1 and Theorem 4.2 can be based on the corresponding proofs in Bacry et al. (2013b), with some modifications to accommodate our shot noise model. In this section, we restate the lemmas and demonstrate the effectiveness of these lemmas on the shot noise model.

Preparations

Proposition 4.2. *Let f be a non-negative measurable function in $L^1(\mathbb{R})$ with $f(t) = 0$ for $t < 0$, given $a > 0$, $h_a(t) = ae^{-at}1_{t\geqslant 0}$*

$$\begin{aligned}\int_0^T (f \star h_a)(t)dt &= \int_0^T \int_0^t af(s)e^{-at}e^{as}\,ds\,dt \\ &= \int_0^T f(s)\,ds - e^{-aT}\int_0^T f(s)e^{as}\,ds \\ &\longrightarrow \int_0^\infty f(t)dt \ \textit{when}\ T \to \infty\end{aligned}$$

Proof. By L'Hôpital's rule, $\lim_{T\to\infty} e^{-aT}\int_0^T f(s)e^{as}\,ds = \lim_{T\to\infty} f(T) = 0$ □

Proposition 4.3. *Let f be an even function which is bounded. Given a positive a, $h_a(t) = ae^{-at}1_{t\geqslant 0}$. In this case, $f(t)h_a(t)$ is integrable over $\mathbb{R}$. We denote the integral of $f(t)h_a(t)$ over $\mathbb{R}$ as*

$$\langle f, h_a\rangle = \int_{\mathbb{R}} f(t)h_a(t)dt$$

Then we have

$$\langle f \star h_a, h_b\rangle = \frac{b}{a+b}\langle f, h_a\rangle + \frac{a}{a+b}\langle f, h_b\rangle$$

Proof.

$$\begin{aligned}
\langle f \star h_a, h_b\rangle &= \int_0^\infty \int_0^\infty f(t-s)h_a(s)h_b(t)\,ds\,dt \\
&= \int_0^\infty \left(\int_0^t f(t-s)h_a(s)\,ds + \int_t^\infty f(s-t)h_a(s)\,ds\right) h_b(t)\,dt \\
&= \int_0^\infty \left(\int_0^t ae^{-at}e^{au}f(u)\,du + \int_t^\infty ae^{-at}e^{-au}f(u)ds\right)\cdot be^{-bt}\,dt \\
&= ab\int_0^\infty f(u)e^{au}\int_u^\infty e^{-(a+b)t}\,dt\,du + ab\int_0^\infty f(u)e^{-au}\int_0^\infty e^{-(a+b)t}\,dt\,du \\
&= \frac{ab}{a+b}\left(\int_0^\infty f(u)e^{-bu}du + \int_0^\infty f(u)e^{-au}\,du\right) \\
&= \frac{b}{a+b}\langle f, h_a\rangle + \frac{a}{a+b}\langle f, h_b\rangle
\end{aligned}$$

□

Proposition 4.4. *Let f be an even function which is bounded. Given two positive values a, b, $h_a(t) = ae^{-at}1_{t\geqslant 0}$ and $h_b(t) = be^{-bt}1_{t\geqslant 0}$. If g_1 and g_2 are two integrable functions on $\mathbb{R}^+$. We have*

$$\langle f \star g_1 \star h_a, g_2 \star h_b\rangle = \frac{b}{a+b}\langle f \star g_1, g_2 \star h_a\rangle + \frac{b}{a+b}\langle f \star g_2, g_1 \star h_b\rangle$$

In particular, when $a = b$, for a matrix of integrable functions G,

$$\langle f \star h_a \star G, h_a \star G\rangle = \frac{1}{2}\langle f \star G, h_a \star G\rangle + \frac{1}{2}\langle G \star h_a, f \star G\rangle$$

Proof. For two positive values u, r, let us define

$$\begin{aligned}
T_1(u,r) &= \int_r^\infty \int_u^\infty f(t-s)e^{-a(s-u)}e^{-b(t-r)}\,ds\,dt \\
&= \int_0^\infty \int_0^\infty f(t+r-s-u)e^{-as}e^{-bt}\,ds\,dt \\
&= \int_0^\infty \int_{r-s-u}^\infty f(y)e^{-as}e^{-b(y-r+s+u)}\,dy\,ds \\
&= \int_{-\infty}^\infty f(y)\left(\int_{(r-y-u)\vee 0}^\infty e^{-(a+b)s}ds\right)e^{-b(y-r+u)}\,dy \\
&= \frac{1}{a+b}\int_{-\infty}^\infty f(y)\left(e^{-a(-y+r-u)}1_{y<r-u} + e^{-b(y-r+u)}1_{y>r-u}\right)\,dy \\
&\overset{f \text{ is even}}{=} \frac{1}{a+b}\left(\int_{u-r}^\infty f(y)e^{-a(y+r-u)}\,dy + \int_{r-u}^\infty f(y)e^{-b(y-r+u)}\,dy\right) \\
&= \frac{1}{a+b}\left(\int_u^\infty f(y-r)e^{-a(y-u)}\,dy + \int_r^\infty f(y-u)e^{-b(y-r)}\,dy\right)
\end{aligned}$$

Now let us finish the proof

$$
\begin{aligned}
&\langle f \star g_1 \star h_a, g_2 \star h_b \rangle \\
=&\int_{\mathbb{R}_+}\int_{\mathbb{R}_+} f(t-s)\Big(h_a(s) \star g_1(s)\Big)\Big(h_b(t) \star g_2(t)\Big)\, ds\, dt \\
=&ab\int_{\mathbb{R}_+}\int_{\mathbb{R}_+} f(t-s)\int_0^s e^{-a(s-u)}g_1(u)\, du \int_0^t e^{-b(t-r)}g_2(r)\, du\, dr\, ds\, dt \\
=&ab\int_{\mathbb{R}_+}\int_{\mathbb{R}_+} T_1(u,r)g_1(u)g_2(r)\, du\, dr \\
=&\frac{ab}{a+b}\int_{\mathbb{R}_+}\int_{\mathbb{R}_+}\left(\int_r^\infty f(y-u)e^{-a(y-r)}\, dy + \int_u^\infty f(y-r)e^{-b(y-u)}\, dy\right) g_1(u)g_2(r)\, du\, dr \\
\overset{(\star)}{=}&\frac{b}{a+b}\langle f \star g_1, g_2 \star h_a\rangle + \frac{a}{a+b}\langle f \star g_2, g_1 \star h_b\rangle
\end{aligned}
$$

$(\star)$ is true due to the following equations

$$
\begin{aligned}
\int_{\mathbb{R}_+} dr \int_{\mathbb{R}_+} du \int_r^\infty dy\, af(y-u)e^{-a(y-r)}g_1(u)g_2(r) &= \int_{\mathbb{R}_+} dr \int_{\mathbb{R}_+} dy \int_0^y du\, af(y-u)e^{-a(y-r)}g_1(u)g_2(r) \\
&= \int_{\mathbb{R}_+} du \int_{\mathbb{R}_+} dy\, af(y-u)\left(g_2(y) \star e^{-ay}\right) g_1(u) \\
&= \langle f \star g_1, g_2 \star h_a\rangle \\
\int_{\mathbb{R}_+} dr \int_{\mathbb{R}_+} du \int_u^\infty dy\, bf(y-r)e^{-b(y-u)}g_1(u)g_2(r) &= \int_{\mathbb{R}_+} dr \int_{\mathbb{R}_+} dy\, bf(y-r)\left(g_1(y) \star e^{-by}\right) g_2(r) \\
&= \langle f \star g_2, g_1 \star h_b\rangle
\end{aligned}
$$

□

Lemma 1 (Lemma 2 in Bacry et al. (2013b))**.** *For all finite stopping times S, one has*

$$
\begin{aligned}
&\mathbb{E}[N_S] = \mu\mathbb{E}[S] + \mathbb{E}[\int_0^S \phi(t-s)N_t\, dt] \\
&\mathbb{E}[N_S] \leqslant R\mu\mathbb{E}[S]
\end{aligned}
\tag{4.A.1}
$$

Proof. The first equation in Eq(4.A.1) is not affected by this kernel form. Therefore we only need to check the second inequality.

- $i = 3$ $\mathbb{E}[N_{3,S_p}] = \mu_3\mathbb{E}[S_p]$ as N_3 is a Poisson process
- $i \in \{4,5\}$, N_4 and N_5 are both delayed processes of N_3 $\mathbb{E}[N_{i,S_p}] \leqslant \mathbb{E}[N_{3,S_p}] = \mu_3\mathbb{E}[S_p]$

- $i \in \{1,2\}$

$$\begin{aligned}
\mathbb{E}[N_{i,S_p}] &= \mu_i \mathbb{E}[S_p] + \mathbb{E}\Big[\int_0^{S_p} \sum_{j=1}^{5} \varphi^{ij}(S_p - t) N_{j,t}\, dt\Big] \\
&= \mu_i \mathbb{E}[S_p] + \mathbb{E}\Big[\int_0^{S_p} \sum_{j=1,2} \varphi^{ij}(S_p - t)(N_{j,t} + N_{j+3,t})\, dt\Big] \\
&\leqslant \mu_i \mathbb{E}[S_p] + \mathbb{E}\Big[(N_{j,S_p} + N_{j+3,S_p}) \int_0^{\infty} \sum_{j=1,2} \varphi^{ij}(S_p - t)\, dt\Big] \\
&\leqslant \mu_i \mathbb{E}[S_p] + \sum_{j=1,2} \\
varphi^{ij} &\|\mathbb{E}[N_{j,S_p} + N_{j+3,S_p}] \\
&\leqslant \mu_i \mathbb{E}[S_p] + \sum_{j=1,2} \|\varphi^{ij}\| \mathbb{E}[N_{j,S_p}] + \mu_3 \mathbb{E}[S_p] \sum_{j=1,2} \|\varphi^{ij}\|
\end{aligned} \tag{4.A.2}$$

Let denote $N_H = \begin{pmatrix} N_1 \\ N_2 \end{pmatrix}$, $\mu_H = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$

$$\begin{aligned}
\mathbb{E}[N_{H,S_p}] &\leqslant \Big(\mu_H + \mu_3 G_H \mathbb{I}\Big)\mathbb{E}[S_p] + G_H \mathbb{E}[N_{H,S_p}] \\
&\leqslant \cdots \\
&\leqslant \Big[\Big(\sum_{k=0}^{n} G_H^k\Big)\mu_H + \mu_3 G_H \Big(\sum_{k=0}^{n} G_H^k\Big)\mathbb{I}_{2\times 1}\Big]\mathbb{E}[S_p] + G_H^{n+1}\mathbb{E}[N_{H,S_p}] \\
&\xrightarrow{n\to\infty} \Big[(I - G_H)^{-1}\mu_H + G_H(I - G_H)^{-1}\mu_3\Big]\mathbb{E}[S_p] \\
&= \begin{pmatrix} \boldsymbol{R}_H & (\boldsymbol{R}_H - I)\mathbb{I}_{2\times 1} \end{pmatrix} \begin{pmatrix} \mu^1 \\ \mu^2 \\ \mu^3 \end{pmatrix} \mathbb{E}[S_p]
\end{aligned} \tag{4.A.3}$$

In conclusion, when $p \to \infty$, $\mathbb{E}[N_S] \leqslant \boldsymbol{R}\mu\mathbb{E}[S]$ □

Lemma 2 (Lemma 3 in Bacry et al. (2013b))**.** *Let h be a Borel and locally bounded function from $\mathbb{R}^+$ to $\mathbb{R}^d$. Then there exists a unique locally bounded function $f : \mathbb{R}^+ \to \mathbb{R}^d$ solution to*

$$f(t) = h(t) + \int_0^t \Phi(t - s) f(s)\, ds$$

given by

$$f_h(t) = h(t) + \int_0^t \Psi(t - s) h(s)\, ds$$

Proof. We can easily follow the proof in Bacry et al. (2013b) until $f_h(t) - f(t) = \int_0^t \Phi(t-s)(f_h(s) - f(s))\, ds$.

- $i = 3$, by the definition of the kernel matrix $\boldsymbol{\Phi}$, we have directly $f_{h,3}(t) - f_3(t) = 0$

- $i \in \{4,5\}$, $f_{h,i}(t) - f_i(t) = -a_{i-3}\int_0^t (f_{h,i}(s) - f_i(s))\,ds \Rightarrow f_{h,i}(t) = f_i(t)$
- $i \in \{1,2\}$,
  $f_{h,i}(t) - f_i(t) = \sum_{j=1}^5 \int_0^t \Phi^{ij}(t-s)(\underbrace{f_{h,i}(s) - f_i(s)}_{=0 \text{ if j=3,4,5}})\,ds = \sum_{j=1}^2 \int_0^t \phi_H^{ij}(t-s)(f_{h,i}(s) - f_i(s))\,ds$

  Let $g_i = |f_{h,i} - f_i|$, and $g = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$, then we have $g(t) \leqslant \int_0^t \phi_H(t-s)g(s)\,ds$. As $\rho(G_H) < 1$, we return to the same proof as in Bacry et al. (2013b).

□

Lemma 3 (Lemma 4 in Bacry et al. (2013b)). *Define the d-dimensional martingale $(M_t)_{t\geqslant 0}$ by*

$$M_t = N_t - \int_0^t \lambda_s\,ds$$

For all $t \in \mathbb{R}^+$, the following equations hold

$$\mathbb{E}[N_t] = t\mu + \left(\int_0^t \Psi(t-s)s\,ds\right)\mu \tag{4.A.4}$$

$$N_t - \mathbb{E}[N_t] = M_t + \int_0^t \Psi(t-s)M_s\,ds \tag{4.A.5}$$

Proof of Theorem 4.1

Lemma 4 (Lemma 5 in Bacry et al. (2013b)). *Let $p \in [0,1]$ and assume that $\int_0^\infty t^p\phi_H(t)\,dt < \infty$ componentwise. Then we have the following properties:*

- *if $p < 1$, we have $T^p(\frac{1}{T}\mathbb{E}[N_{Tu}] - uR\mu) \to 0$ as $T \to \infty$ uniformly for $u \in [0,1]$*
- *if $p = 1$, we have $T(\frac{1}{T}\mathbb{E}[N_T] - R\mu) \to B\mu$ as $T \to \infty$ (see (4.A.7) for the definition of B)*

Proof. $\int_0^\infty t^p\phi_H(t)\,dt < \infty \Rightarrow \int_0^\infty t^p\psi_H(t)\,dt < \infty$ is proven in Bacry et al. (2013b). We need to further prove that $\int_0^\infty t^p|\Psi(t)|\,dt < \infty$ componentwise as well.

In fact, it is enough to prove that

- $\int_0^\infty t^p\psi_H(t) \star (e^{\Gamma t})\,dt < \infty$
- $\int_0^\infty t^p e^{\Gamma t}\,dt < \infty$

Since $\Gamma \leqslant 0$ componentwise, the second inequality can be deduced straightforwardly. In order to

establish the first inequality, we must prove that $\int_0^\infty t^p(\psi_H(t) \star e^{-at})\,dt < \infty$ holds true for $a > 0$.

$$\begin{aligned}
\int_1^\infty t^p(\psi_H(t) \star e^{-at})\,dt &= \int_1^\infty t^p \int_0^t \psi_H(s)e^{-a(t-s)}ds\,dt \\
&= \int_0^\infty \psi_H(s)e^{as}\Big(\int_{s\vee 1}^\infty t^p e^{-at}\,dt\Big)\,ds \\
&= \int_0^\infty \psi_H(s)e^{as}\Big(\frac{1}{a}e^{-a(s\vee 1)}(s\vee 1)^p + \frac{p}{a}\int_{s\vee 1}^\infty e^{-at}t^{p-1}\,dt\Big)ds \\
&\leqslant \frac{1}{a}\int_0^\infty \psi_H(s)(s\vee 1)^p\,ds + \frac{p}{a}\int_0^\infty \psi_H(s)e^{as}\int_{s\vee 1}^\infty e^{-at}t^{p-1}\,dt\,ds \\
&\leqslant \frac{1}{a}\int_0^\infty \psi_H(s)(s\vee 1)^p\,ds + \frac{p}{a^2}\int_0^\infty \psi_H(s)e^{as}e^{-a(s\vee 1)}(s\vee 1)^{p-1}\,ds \\
&\leqslant \frac{1}{a}\int_0^1 \psi_H(s)\,ds + \frac{1}{a}\int_1^\infty \psi_H(s)s^p\,ds + \frac{p}{a^2}\int_0^\infty \psi_H(s)(s\vee 1)^p\,ds < \infty
\end{aligned} \tag{4.A.6}$$

As $\psi_H(t) \star (e^{\Gamma t})$ is a continuous function, $\int_0^1 t^p(\psi_H(t) \star e^{-at})\,dt < \infty$.

When $p = 1$, from Bacry et al. (2013b), we have $\int_0^\infty t\psi_H(t)\,dt = (I_2 - G_H)^{-1}\Big(\int_0^\infty t\phi_H(t)\,dt\Big)(I_2 - G_H)^{-1} =: B_H$

$$\int_0^\infty t\Psi(t)\,dt = \begin{pmatrix} B_H & B_H\mathbb{I}_{2\times 1} + \|\psi_H\|\frac{1}{\gamma} & \|\psi_H\|\frac{1}{\Gamma} \\ 0_{1\times 2} & 0 & 0_{1\times 2} \\ 0_{2\times 2} & \frac{1}{\gamma} & \frac{1}{\Gamma} \end{pmatrix} =: B \tag{4.A.7}$$

where $\frac{1}{\gamma} = \begin{pmatrix} \frac{1}{a_1} \\ \frac{1}{a_2} \end{pmatrix}$ and $\frac{1}{\Gamma} = \begin{pmatrix} -\frac{1}{a_1} & 0 \\ 0 & -\frac{1}{a_2} \end{pmatrix}$ □

Lemma 5 (Lemma 6 in Bacry et al. (2013b))**.** *There exists a constant C_{μ,ϕ_H} such that for all $t, \Delta \geqslant 0$,*

$$\mathbb{E}\left(\sup_{t\leqslant s\leqslant t+\Delta} \|M_s - M_t\|^2\right) \leqslant C_{\mu,\phi_H}\Delta$$

Proof. ▫ $i = 3$, $\mathbb{E}[N_3(t+\Delta) - N_3(t)] = \mu_3\Delta$

▫ $i \in \{4, 5\}$, From (4.A.4)

$$\begin{aligned}
\mathbb{E}[N_i(t+\Delta) - N_i(t)] &\overset{a_i' = a_{i-3}}{=} \left(\int_0^{t+\Delta} a_i' s e^{-a_i'(t+\Delta-s)}ds - \int_0^t a_i' s e^{-a_i'(t-s)}ds\right)\mu_3 \\
&\leqslant \Delta\left(\int_0^\infty a_i' e^{-a_i' t}\,dt\right)\mu_3 = \mu_3\Delta
\end{aligned} \tag{4.A.8}$$

▫ $i \in \{1, 2\}$, Let denote $N_H = \begin{pmatrix} N_1 \\ N_2 \end{pmatrix}$, $\mu_H = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$

$$\mathbb{E}[N_{H,t}] = t\mu_H + \left(\int_0^t \psi_H(t-s)s\,ds\right)\mu_H + \left(\int_0^t s(\psi_H(s) \star \bar{\gamma})\,ds\right)\mu_3$$

Since $\int_0^\infty \psi_H(s) \star \bar{\gamma} ds \overset{\text{(By Proposition 4.2)}}{=} \int_0^\infty \psi_H(s) \mathbb{I}_{2\times 1} ds = \left((I_2 - G_H)^{-1} - I_2\right) \mathbb{I}_{2\times 1}$, we obtain

$$\mathbb{E}[N_H(t+\Delta) - N_H(t)] \leqslant \Delta (I_2 - G_H)^{-1} \mu_H + \Delta \left((I_2 - G_H)^{-1} - I_2\right) \mathbb{I}_{2\times 1} \mu_3 \tag{4.A.9}$$

In summary, we obtain $\mathbb{E}[N_{t+\Delta} - N_t] \leqslant \Delta \boldsymbol{R} \mu$ componentwise. □

Proof of Theorem 4.3

Proof. From Bacry et al. (2013b) Theorem 3, we have $V_{\Delta_T,T}(X) - \mathbf{v}_{\Delta_T} \to 0$ when $T \to \infty$, where

$$\mathbf{v}_\Delta = \int_{\mathbb{R}_+^2} \left(1 - \frac{|t-s|}{\Delta}\right)^+ R(s) \Sigma R(t)^\top \, ds \, dt$$

Let us denote by $f_\Delta(t)$ the function $\left(1 - \frac{|t|}{\Delta}\right)^+$ which is even bounded on $\mathbb{R}$.

Then the covariance matrix for $\bar{N} = \begin{pmatrix} N_1 + N_4 \\ N_2 + N_5 \end{pmatrix}$ is

$$\begin{aligned}
c_\Delta =& \frac{1}{\Delta} Cov(\bar{N}_{t+\Delta} - \bar{N}_t, \bar{N}_{t+\Delta} - \bar{N}_t) \\
=& \begin{pmatrix} I_2 & 0_{2\times 1} & I_2 \end{pmatrix} \mathbf{v}_\Delta \begin{pmatrix} I_2 & 0_{2\times 1} & I_2 \end{pmatrix}^\top \\
=& \int_{\mathbb{R}_+^2} f_\Delta(t-s) \begin{pmatrix} I_2 & 0_{2\times 1} & I_2 \end{pmatrix} R(s) \Sigma R(t)^\top \begin{pmatrix} I_2 & 0_{2\times 1} & I_2 \end{pmatrix}^\top ds \, dt \\
\overset{(\clubsuit)}{=}& \int_{\mathbb{R}_+^2} f_\Delta(t-s) R_H(s) \Sigma_H R_H(t)^\top \, ds \, dt + \mu_3 \int_{\mathbb{R}_+^2} f_\Delta(t-s) \Big(R_H(s) \star \bar{\gamma}(s)\Big) \Big(R_H(t) \star \bar{\gamma}(t)\Big)^\top ds \, dt \\
& + \mu_3 \int_{\mathbb{R}_+^2} f_\Delta(t-s) \Big(R_H(s) + R_H(s) \star \bar{\Gamma}(s)\Big) \Big(R_H(t) + R_H(t) \star \bar{\Gamma}(t)\Big)^\top ds \, dt \\
=& (T_1) + (T_2)
\end{aligned}$$

where

- The first term

$$(T_1) = \int_{\mathbb{R}_+^2} f_\Delta(t-s) R_H(s) \bar{\Sigma} R_H(t)^\top \, ds \, dt$$

with $\bar{\Sigma} = \Sigma_H + \mu_3 I_2$ and $(\clubsuit)$ is from

$$\begin{aligned}
\begin{pmatrix} I_2 & 0_{2\times 1} & I_2 \end{pmatrix} R(s) &= \begin{pmatrix} R_H & \psi_H \star \bar{\gamma} + \bar{\gamma} & \psi_H + \psi_H \star \bar{\Gamma} + \delta I_2 + \bar{\gamma} \end{pmatrix} \\
&= \begin{pmatrix} R_H & R_H \star \bar{\gamma} & R_H + R_H \star \bar{\Gamma} \end{pmatrix}
\end{aligned}$$

▫ The second term is

$$\begin{aligned}(T_2) =&\mu_3 \int_{\mathbb{R}_+^2} f_\Delta(t-s)\Big(R_H(s)\star\bar\gamma(s)\Big)\Big(R_H(t)\star\bar\gamma(t)\Big)^\top ds\,dt\\ &+\mu_3 \int_{\mathbb{R}_+^2} f_\Delta(t-s)\Big(R_H(s)\star\bar\Gamma(s)\Big)\Big(R_H(t)\star\bar\Gamma(t)\Big)^\top ds\,dt\\ &+\mu_3 \int_{\mathbb{R}_+^2} f_\Delta(t-s)R_H(s)\Big(R_H(t)\star\bar\Gamma(t)\Big)^\top ds\,dt\\ &+\mu_3 \int_{\mathbb{R}_+^2} f_\Delta(t-s)\Big(R_H(s)\star\bar\Gamma(s)\Big)R_H(t)^\top ds\,dt\\ =&\mu_3 \int_{\mathbb{R}_+^2} f_\Delta(t-s)\Big(R_H(s)\star\bar\Gamma(s)\Big)\begin{pmatrix}0&1\\1&0\end{pmatrix}\Big(R_H(t)\star\bar\Gamma(t)\Big)^\top ds\,dt\end{aligned}$$

The last equality is due to Proposition 4.4.

□

Proposition 4.5. *Consider two vectors of bounded and integrable functions f, g on $\mathbb{R}^+$,*

$$\langle(1-\tfrac{|\cdot|}{\Delta})^+, f\rangle = \int_0^\Delta (1-\frac{t}{\Delta})f(t)dt = \frac{1}{2}f(0)\Delta + o(\Delta) \tag{4.A.10}$$

$$\langle f, (1-\tfrac{|\cdot|}{\Delta})^+ \star g\rangle = \int_{\mathbb{R}_+^2} (1-\frac{|t-s|}{\Delta})^+ f(t)g(s)\,ds\,dt = \Delta\langle f,g\rangle + o(\Delta) \tag{4.A.11}$$

Proof. Let us prove the second equation.

$\left(1-\frac{|t-s|}{\Delta}\right)^+$ is non-zero when $t<s<t+\Delta$ or $s<t<s+\Delta$,

$$\begin{aligned}&\int_{\mathbb{R}_+^2} (1-\frac{|t-s|}{\Delta})^+ f(s)g(t)\,ds\,dt\\ =&\int_{\mathbb{R}^+}\int_t^{t+\Delta}(1-\frac{s-t}{\Delta})f(s)g(t)^\top ds\,dt + \int_{\mathbb{R}^+}\int_s^{s+\Delta}(1-\frac{t-s}{\Delta})f(s)g(t)^\top dt\,ds\\ =&\int_{\mathbb{R}^+}\left(\int_0^\Delta(1-\frac{u}{\Delta})f(t+u)\,du\right)g(t)^\top dt + \int_{\mathbb{R}^+} f(s)\left(\int_0^\Delta(1-\frac{u}{\Delta})g(s+u)^\top du\right)ds\\ \overset{(4.A.10)}{=}&\Delta\int_{\mathbb{R}^+} f(t)g(t)^\top dt + o(\Delta) = \Delta\langle f,g\rangle + o(\Delta)\end{aligned}$$

□

Proof of Corollary 6. Following the proof for Theorem 4.3,

$$\begin{aligned}(T_1) =& \int_{\mathbb{R}_+^2}\left(1-\frac{|t-s|}{\Delta}\right)^+ (I_2\delta(s)+\psi_H(s))\bar\Sigma(I_2\delta(t)+\psi_H(t))^\top ds\,dt\\ =&\bar\Sigma + \underbrace{\int_0^\Delta\left(1-\frac{t}{\Delta}\right)\bar\Sigma\psi_H(t)^\top dt}_{(T_{1,2})} + \underbrace{\int_0^\Delta\left(1-\frac{t}{\Delta}\right)\psi_H(t)\bar\Sigma\,dt}_{(T_{1,3})}\\ &+\underbrace{\int_{\mathbb{R}_+^2}\left(1-\frac{|t-s|}{\Delta}\right)^+\psi_H(s)\bar\Sigma\psi_H(t)^\top ds\,dt}_{(T_{1,4})}\end{aligned}$$

- $(T_{1,2}) = \frac{1}{2}\bar{\Sigma}\psi_H(0)^\top\Delta + o(\Delta)$ and $(T_{1,3}) = \frac{1}{2}\psi_H(0)\bar{\Sigma}\Delta + o(\Delta)$ by Eq.(4.A.10) of Proposition 4.5.
- $(T_{1,4}) = \Delta\int_{\mathbb{R}^+}\psi_H(t)\bar{\Sigma}\psi_H(t)^\top dt + o(\Delta)$ directly from Eq.(4.A.11) of Proposition 4.5

Therefore $(T_1) = \bar{\Sigma} + \left(\frac{1}{2}\bar{\Sigma}\psi_H(0)^\top + \frac{1}{2}\psi_H(0)\bar{\Sigma} + \int_{\mathbb{R}^+}\psi_H(t)\bar{\Sigma}\psi_H(t)^\top dt\right)\Delta + o(\Delta)$

For (T_2), let us denote $(\psi_H \star \bar{\Gamma})$ by $\psi_{H,\Gamma}$, we have

$$(T_2) = \mu_3\int_{\mathbb{R}_+^2}\left(1-\frac{|t-s|}{\Delta}\right)^+\left(\bar{\Gamma}(s)+\psi_H(s)\star\bar{\Gamma}(s)\right)\begin{pmatrix}0&1\\1&0\end{pmatrix}\left(\bar{\Gamma}(t)+\psi_H(t)\star\bar{\Gamma}(t)\right)^\top ds\,dt$$

$$= (T_{2,1}) + (T_{2,2}) + (T_{2,3}) + (T_{2,4})$$

where

- By Proposition 4.3 and $H_a\left(\left(1-\frac{|t|}{\Delta}\right)^+\right) = \int_0^\infty\left(1-\frac{|t|}{\Delta}\right)^+ ae^{-at}dt = 1-\frac{1-e^{-a\Delta}}{a\Delta}$

$$(T_{2,1}) = \mu_3\int_{\mathbb{R}^+}\left(1-\frac{|t-s|}{\Delta}\right)^+\bar{\Gamma}(s)\begin{pmatrix}0&1\\1&0\end{pmatrix}\bar{\Gamma}(t)^\top ds\,dt$$

$$= \mu_3\int_{\mathbb{R}^+}\left(1-\frac{|t-s|}{\Delta}\right)^+ a_1a_2\begin{pmatrix}0&e^{-a_1s}e^{-a_2t}\\e^{-a_2s}e^{-a_1t}&0\end{pmatrix}ds\,dt$$

$$= \mu_3\left(1-\frac{a_2}{a_1+a_2}\frac{1-e^{-a_1\Delta}}{a_1\Delta}-\frac{a_1}{a_1+a_2}\frac{1-e^{-a_2\Delta}}{a_2\Delta}\right)\begin{pmatrix}0&1\\1&0\end{pmatrix}$$

$$= \mu_3\frac{a_1a_2}{a_1+a_2}\Delta\begin{pmatrix}0&1\\1&0\end{pmatrix} + o(\Delta)$$

- Similarly,

$$(T_{2,2}) = \mu_3\int_{\mathbb{R}_+^2}\left(1-\frac{|t-s|}{\Delta}\right)^+\psi_H(s)\star\bar{\Gamma}(s)\begin{pmatrix}0&1\\1&0\end{pmatrix}\bar{\Gamma}(t)^\top ds\,dt$$

$$\overset{\text{By Prop 4.5}}{=} \mu_3\Delta\int_{\mathbb{R}^+}\psi_H(t)\star\bar{\Gamma}(t)\begin{pmatrix}0&1\\1&0\end{pmatrix}\bar{\Gamma}(t)^\top dt + o(\Delta)$$

$$(T_{2,3}) = \mu_3\int_{\mathbb{R}_+^2}\left(1-\frac{|t-s|}{\Delta}\right)^+\bar{\Gamma}(s)\begin{pmatrix}0&1\\1&0\end{pmatrix}\left(\psi_H(t)\star\bar{\Gamma}(t)\right)^\top ds\,dt$$

$$= \mu_3\Delta\int_{\mathbb{R}^+}\psi_H(t)\begin{pmatrix}0&1\\1&0\end{pmatrix}\left(\psi_H(t)\star\bar{\Gamma}(t)\right)^\top dt + o(\Delta)$$

$$(T_{2,4}) = \mu_3\int_{\mathbb{R}_+^2}\left(1-\frac{|t-s|}{\Delta}\right)^+\left(\psi_H(s)\star\bar{\Gamma}(s)\right)\begin{pmatrix}0&1\\1&0\end{pmatrix}\left(\psi_H(t)\star\bar{\Gamma}(t)\right)^\top ds\,dt$$

$$= \mu_3\Delta\int_{\mathbb{R}^+}\left(\psi_H(t)\star\bar{\Gamma}(t)\right)\begin{pmatrix}0&1\\1&0\end{pmatrix}\left(\psi_H(t)\star\bar{\Gamma}(t)\right)^\top dt + o(\Delta)$$

When $a \gg 1$, for a function $f \in \mathcal{C}^1(\mathbb{R}^+) \bigcap L^1(\mathbb{R}^+)$, and suppose that $f'(t)$ is bounded on $\mathbb{R}^+$,

$$\int_0^\infty f(t) a e^{-at}\, dt = f(0) + \int_0^\infty f'(t) e^{-at}\, dt = f(0) + O(\frac{1}{a})$$

Therefore, $a_1 \psi_H(t) \star e^{-a_1 t} = \psi_H(t) - \psi_H(0) e^{-a_1 t} + O(\frac{1}{a_1})$, and

$$\int_0^\infty a_1 a_2 e^{-a_2 t} \psi_H(t) \star e^{-a_1 t}\, dt = \psi_H(0) - \frac{a_2}{a_1 + a_2} \psi_H(0) + O(\frac{\Delta}{a_1 \wedge a_2}) = \frac{a_1}{a_1 + a_2} \psi_H(0) + O(\frac{\Delta}{a_1 \wedge a_2})$$

Now let us consider in particular $\underline{a_1 = a_2 = a}$

$$(T_{2,1}) = \mu_3 \left(1 - \frac{1 - e^{-a\Delta}}{a\Delta}\right) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$(T_{2,2}) = \frac{\Delta}{2} \mu_3 \psi_H(0) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + O(\frac{\Delta}{a})$$

$$(T_{2,3}) = \frac{\Delta}{2} \mu_3 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \psi_H(0)^\top + O(\frac{\Delta}{a})$$

$$(T_{2,4}) = \mu_3 \Delta \int_0^\infty \psi_H(t) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \psi_H(t)^\top dt + O(\frac{\Delta}{a})$$

In summary, for $\Delta \sim O(\frac{1}{a})$

$$\begin{aligned}
c_\Delta =& (T_1) + (T_2) \\
=& \bar{\Sigma} + \left(\frac{1}{2}\bar{\Sigma}\psi_H(0)^\top + \frac{1}{2}\psi_H(0)\bar{\Sigma} + \int_{\mathbb{R}^+} \psi_H(t)\bar{\Sigma}\psi_H(t)^\top dt\right)\Delta + \mu_3 \left(1 - \frac{1 - e^{-a\Delta}}{a\Delta}\right) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
&+ \frac{1}{2}\psi_H(0) \begin{pmatrix} 0 & \mu_3 \\ \mu_3 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & \mu_3 \\ \mu_3 & 0 \end{pmatrix} \psi_H(0)^\top + \Delta \int_0^\infty \psi_H(t) \begin{pmatrix} 0 & \mu_3 \\ \mu_3 & 0 \end{pmatrix} \psi_H(t)^\top dt + o(\Delta) \\
=& \begin{pmatrix} \Lambda_{P,1} & 0 \\ 0 & \Lambda_{P,2} \end{pmatrix} + \left(1 - \frac{1 - e^{-a\Delta}}{a\Delta}\right) \begin{pmatrix} 0 & \mu_3 \\ \mu_3 & 0 \end{pmatrix} \\
&+ \Delta\Big(\frac{1}{2}\psi_H(0) \begin{pmatrix} \Lambda_{P,1} & \mu_3 \\ \mu_3 & \Lambda_{P,2} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \Lambda_{P,1} & \mu_3 \\ \mu_3 & \Lambda_{P,2} \end{pmatrix} \psi_H(0)^\top + \int_0^\infty \psi_H(t) \begin{pmatrix} \Lambda_{P,1} & \mu_3 \\ \mu_3 & \Lambda_{P,2} \end{pmatrix} \psi_H(t)^\top dt\Big) \\
&+ o(\Delta)
\end{aligned}$$

□

Proof of Corollary 5. Follow the proof for Theorem 4.3,

$$\lim_{\Delta \to \infty} (T_1) = \int_{\mathbb{R}_+^2} R_H(s) \bar{\Sigma} R_H(t)^\top\, ds\, dt = \boldsymbol{R}_H \bar{\Sigma} \boldsymbol{R}_H^\top$$

And by Proposition 4.2,

$$\lim_{\Delta\to\infty}(T_2) = \mu_3 \int_{\mathbb{R}^+} R_H(s) \star \bar{\Gamma}(s)\, ds \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \int_{\mathbb{R}^+} (R_H(t) \star \bar{\Gamma}(t))^\top\, dt$$

$$= \mu_3 \boldsymbol{R}_H \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \boldsymbol{R}_H^\top$$

□

4.B - Simulation (latent-behavior model)

In this section, we will introduce two different techniques to simulate a $(2 \times 2 + 1)$-dimensional Hawkes process with shot noise (i.e., Model (4.2.4)).

Suppose that the Hawkes kernels are some sum of exponential functions as follows :

$$\phi_{ij}(t) = \sum_{u=1}^{U} \alpha_{u,ij}\beta_u exp(-\beta_u t)$$

We first provide the simulating algorithm 2 for a bivariate Poisson processes, i.e., two delayed shot noise processes.

Algorithm 2: Generate bivariate Poisson noises

Input: a_1, a_2, μ_3, T

On $[0, T]$, generate a poisson process [2] $N_3 \sim PP(\mu_X)$, $N_3 = \{t_k\}_{k=1}^M$;

Generate the two sets of delays $(\Delta_k^{(i)})_{k=1}^M \sim_{iid} \mathcal{E}xp(a_i)$, for $i = 1, 2$;

$N_4 \leftarrow \{t_k + \Delta_k^{(1)}\}_{k=1}^M$;

$N_5 \leftarrow \{t_k + \Delta_k^{(2)}\}_{k=1}^M$;

$\mathcal{N} = \bigcup_{i=4,5} \{(t_k, i) | t_k \in N_i \cap [0, T]\}$;

return $\mathcal{N}$ *(ascending ordered by the first elements)*

Simulation algorithms

We propose two algorithms to simulate the Hawkes process with shot noise: the cluster algorithm and the thinning algorithm.

The cluster algorithm generates a Hawkes process recursively by creating clusters of immigrants. In this model, the immigrants consist of the superposition of immigrants from self-exciting processes and from shot noise. The generation of immigrants is accomplished through Algorithm 2. Once the immigrants have been generated, the clustering approach is the same as for classical Hawkes processes Jovanović et al. (2015). The process is described in detail in Algorithm 3.

The thinning algorithm is based on the Ogata algorithm Ogata (1981). Algorithm 4 shows the procedure for generating a Hawkes process with shot noise.

Algorithm 3: Cluster algorithm : generate Hawkes processes with shot noise

Setting : Hawkes kernels $\phi_{ij}(t) = \sum_{u=1}^{U} \alpha_{u,ij}\beta_u exp(-\beta_u t)$
Input: μ_1, μ_2, $\{(\alpha_u)_{2\times 2}\}_{u=1}^{U}$, $\{\beta_u\}_{u=1}^{U}$, μ_3, a_1, a_2, T;
for $i \in \{1,2\}$ **do**
  Generate N_i immigrant process, which is a poisson process on $[0,T]$, $N_i \sim PP(\mu_i)$
Using Algorithm 2, generate the shot noise (the "synchronized" immigrants) N_4 and N_5;
for $i \in \{1,2\}$ **do**
  $\bar{N}_i \leftarrow sorted(N_i \cup N_{i+3})$;
  for $t \in \bar{N}_i$ **do**
    `/* Generate the cluster of an immigrant */`
    for $j \in \{1,2\}$ **do**
      for $u \in [1:U]$ **do**
        Generate $(p_n) \sim PP(1)$ on $[0,\Delta]$ where $\Delta = \alpha_{u,ji}\big(1 - exp(-\beta_u(T-t))\big)$;
        $t_{j,n} = t - \frac{1}{\beta_u}\log(1 - \frac{p_n}{\alpha_{u,ji}})$;
        $\bar{N}_j \leftarrow \bar{N}_j \bigcup \{t_{j,n}\}_n$;
        Repeat the generation of cluster part on $\{t_{j,n}\}_n$ until it becomes an empty set
return $\{sorted(\bar{N}_i)\}_{i\in\{1,2\}}$

4.C - Sequential Monte Carlo Expectation-Maximization Method

Several related studies have already applied the Expectation-Maximization method (EM) to address similar problems, including Cappé (2009); Linderman et al. (2017); Mei et al. (2019); Shelton et al. (2018). However, these studies are not directly applicable to our model. In this section, we will introduce the EM method for our latent-behavior model. Specifically, we will only consider a simplified version of the model, which is a 2-dimensional process with $a_1 = \infty$ in Eq. (4.2.4). In this case, N_3 and N_4 coincide, and the latent-behavior model can be expressed as follows:

$$\begin{cases} N_1 : \lambda_1(t) = \mu_1 + \int_0^t \varphi_{11}(t-s)\, d(N_1(s) + N_4(s)) + \int_0^t \varphi_{12}(t-s)\, d(N_2(s) + N_5(s)) \\ N_2 : \lambda_2(t) = \mu_2 + \int_0^t \varphi_{22}(t-s)\, d(N_2(s) + N_5(s)) + \int_0^t \varphi_{21}(t-s)\, d(N_1(s) + N_4(s)) \\ N_4 : \lambda_4(t) = \mu_3 \\ N_5 : \lambda_5(t) = a\,(N_4(t) - N_5(t)) \end{cases} \tag{4.C.1}$$

with $\bar{N}_1 = N_1 + N_4$ and $\bar{N}_2 = N_2 + N_5$ being the two observable processes.

Let us reframe the problem. Within the event space $\mathcal{E} = 1, 2, 4, 5$, as illustrated in Figure 4.C.1, these elements are associated with specific symbols: 1 is denoted by a blue diamond (♦), 2 by a blue circle (●), 4 by a red diamond (◊), and 5 by a red circle (○). For an event type $e \in \mathcal{E}$, we denote the shape of e as r_e. If e is diamond ($e = 1$ or 4) then $r_e = 1$; if e is circle, then ($e = 2$ or 5) $r_e = 2$. We also denote the color of e as Z_e. If e is blue ($e = 1$ or 2) then $Z_e = 0$; if e is red ($e = 4$ or 5) then $Z_e = 1$. In other words, $Z = 0$ means the event is from the Hawkes part N_1 or N_2, $Z = 1$ indicates that the event is a shot noise i.e., from N_4 or N_5.

We can represent our shot noise model differently by breaking down each complete event X into

Algorithm 4: Ogata algorithm : generate Hawkes processes with shot noise

Setting : Hawkes kernels $\phi_{ij}(t) = \sum_{u=1}^{U} \alpha_{u,ij}\beta_u exp(-\beta_u t)$
Input: μ_1, μ_2, $\{(\alpha_u)_{2\times 2}\}_{u=1}^{U}$, $\{\beta_u\}_{u=1}^{U}$, μ_3, a_1, a_2, T;
Using Algorithm 2, generate the bivariate shot noise processes N_4 and N_5;
$s \leftarrow 0$;
$\mathcal{H} \leftarrow \varnothing$, $\mathcal{N} \leftarrow sorted\{(\cup_{t\in N_4}(t,1)) \cup (\cup_{t\in N_5}(t,2))\}$;
$(t_{noise}, e_{noise}) \leftarrow \mathcal{N}[1]$; /* the first element in $\mathcal{N}$ */
for $u, i \in \{1,2,\ldots,U\} \times \{1,2\}$ **do**
    $conv_{u,i} \leftarrow 0$; /* the convolution term in intensity */
while $s < T$ **do**
    $\lambda^* \leftarrow \sum_i(\mu_i + \sum_{u=1}^{U} conv_{u,i})$;
    Generate $v \sim \mathcal{U}(0,1)$;
    $w \leftarrow -\log v/\lambda^*$; /* $w \sim \mathcal{E}(\bar{\lambda})$ */
    if $t_{noise} < s + w$ **then**
        /* This part is the only difference from the Ogata Algorithm for the classical Hawkes Model, if the candidate $s+w$ comes after the next noise immigrant */
        For $u, i \in \{1,2,\ldots,U\} \times \{1,2\}$, $conv_{u,i} \leftarrow conv_{u,i}e^{-\beta_u(t_{noise}-s)} + \alpha_{u,i,e_{noise}}$;
        $s \leftarrow t_{noise}$;
        $(t_{noise}, e_{noise}) \leftarrow Next(\mathcal{N})$;
    else
        $s \leftarrow s + w$;
        Generate $D \sim \mathcal{U}(0,1)$;
        /* accepting this candidate with probability $\frac{\sum_{i=1}^{2}(\mu_i + \sum_{u=1}^{U} conv_{u,i}e^{-\beta_u w})}{\lambda^*}$ */
        if $D \leqslant \frac{\sum_{i=1}^{2}(\mu_i + \sum_{u=1}^{U} conv_{u,i}e^{-\beta_u w})}{\lambda^*}$ **then**
            $k \leftarrow 1$;
            while $D > \frac{\sum_{i=1}^{2}(\mu_i + \sum_{u=1}^{U} conv_{u,i}e^{-\beta_u w})}{\lambda^*}$ **do**
                $k \leftarrow k + 1$;
            $\mathcal{H} \leftarrow \mathcal{H} \cup \{(s,k)\}$;
            For $u = 1,2,\ldots,U$ and $i = 1,2$, $conv_{u,i} \leftarrow conv_{u,i}e^{-\beta_u w} + \alpha_{u,i,k}$; /* update $conv$ */
        else
            For $u = 1,2,\ldots,U$ and $i = 1,2$, $conv_{u,i} \leftarrow conv_{u,i}e^{-\beta_u w}$; /* update $conv$ */
return $\mathcal{H} \cup \mathcal{N}$ *(ascending ordered by the first elements)*

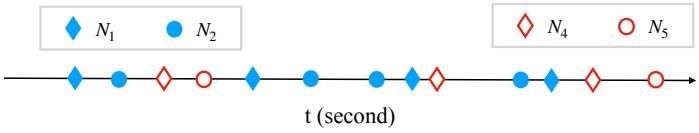


Figure 4.C.1 – Observable : events shape r_e (diamond or circle), unobservable : event color Z_e (blue or red).

three distinct features: the arrival time t, the event shape r_e, and the event color Z_e. This is denoted as $X = (t, e) = (\underbrace{t, r_e}_{Y}, Z_e)$, as shown in the figure above.

- $Y = (t, r_e)$ is an observable feature with the event arrival time t and the event shape $r_e \in \{1, 2\}$ (diamond or circle).
- The color Z_e is unobservable.

Suppose the Hawkes kernels are exponential functions, that is $\varphi_{ij}(t) = \alpha_{ij}\beta e^{-\beta t}$. Let us introduce the EM method for this model, considering a realization with N events $(X_n)_{n=1}^N$, where each event is represented as $X_n = (t_n, e_n)$. We define the sigma-algebra $\mathcal{F}_{t-} = \sigma(X_s, s < t)$, and the observable part of the data is denoted as $Y_n = (t_n, r_n)$. Note that from now we simplify r_{e_n} and Z_{e_n} as r_n and Z_n, respectively.

Likelihood function

Suppose β is known, let θ denote the other parameters of this model, $\theta = (\mu, \alpha, a)$. Define the following global functions, which are **independent of the latent variable** Z:

$$\begin{aligned} g_1(t) &= \sum_{t_k<t, r_k=1} \beta e^{-\beta(t-t_k)} \\ g_2(t) &= \sum_{t_k<t, r_k=2} \beta e^{-\beta(t-t_k)} \\ G_i(t) &= \int_0^t g_i(s)ds \quad \text{for } i \in \{1, 2\} \end{aligned} \tag{4.C.2}$$

Therefore the values of these functions at time t_k are :

$$\begin{aligned} g_i(t_k) &= \sum_{t_l<t_k, r_k=i} \beta e^{-\beta(t_k-t_l)} \\ &= e^{-\beta(t_k-t_{k-1})} \sum_{t_l<t, r_k=i} \beta e^{-\beta(t_{k-1}-t_l)} + \beta e^{-\beta(t_k-t_{k-1})} 1_{r_k=i} \\ &= e^{-\beta(t_k-t_{k-1})} g_i(t_{k-1}) + \beta e^{-\beta(t_k-t_{k-1})} 1_{r_{k-1}=i} \\ H_i(t_k) &= G_i(t_k) - G_i(t_{k-1}) = \int_{t_{k-1}}^{t_k} g_i(s)ds \\ &= \frac{1 - e^{-\beta(t_k-t_{k-1})}}{\beta} g_i(t_{k-1}) + (1 - e^{-\beta(t_k-t_{k-1})}) 1_{r_{k-1}=i} \end{aligned} \tag{4.C.3}$$

The log likelihood function of the complete model is:

$$\begin{aligned}
L(\theta) &= \sum_{i=1,2,4,5} \Big(\int_0^T \log \lambda_i(t)\, dN_i(t) - \int_0^T \lambda_i(t)\, dt \Big) \\
&= \underbrace{\sum_{i=1}^{2} \int_0^T \log \lambda_i(t)\, dN_i(t)}_{I_1} - \underbrace{\sum_{i=1}^{2} \int_0^T \lambda_i(t)\, dt}_{I_2} + \underbrace{\sum_{i=4}^{5} \Big(\int_0^T \log \lambda_i(t)\, dN_i(t) - \int_0^T \lambda_i(t)\, dt \Big)}_{I_3}
\end{aligned}$$

Let us calculate each term separately:

$$\begin{aligned}
I_1 &= \sum_{i=1}^{2} \sum_{t_k, e_k = i} \log \big(\mu_i + \sum_{j=1}^{2} \alpha_{ij} \int_0^{t_k} \beta e^{-\beta(t_k - s)} (dN_j(s) + dN_{j+3}(s)) \big) \\
&= \sum_{i=1}^{2} \sum_{t_k, e_k = i} \log(\mu_i + \sum_{j=1}^{2} \alpha_{ij} \underbrace{\sum_{t_l < t_k, r_l = j} \beta e^{-\beta(t_k - t_l)}}_{g_j(t_k)}) \\
&= \sum_{i=1}^{2} \sum_{t_k, e_k = i} \log(\mu_i + \sum_{j=1}^{2} \alpha_{ij} g_j(t_k))
\end{aligned}$$

$$\begin{aligned}
I_2 &= \sum_{i=1}^{2} \int_0^T \Big(\mu_i + \sum_{j=1}^{2} \alpha_{ij} \int_0^t \beta e^{-\beta(t-s)} (dN_j(s) + dN_{j+3}(s)) \Big)\, dt \\
&= (\mu_1 + \mu_2) T + \sum_{i=1}^{2} \sum_{\substack{t_k \\ t_0 = 0 \\ t_{N+1} = T}} \sum_{j=1}^{2} \int_{t_{k-1}}^{t_k} \underbrace{\int_0^t \beta e^{-\beta(t_k - s)} (dN_j(s) + dN_{j+3}(s))}_{g_j(t)}\, dt \\
&= (\mu_1 + \mu_2) T + \sum_{i=1}^{2} \sum_{t_k} \sum_{j=1}^{2} \alpha_{ij} H_j(t_k) - \sum_{i=1}^{2} \sum_{j=1}^{2} \alpha_{ij} H_j(T)
\end{aligned}$$

$$\begin{aligned}
I_3 = &\sum_{t_k, e_k = 3} \log(\mu_3) + \sum_{t_k, e_k = 4} 1_{N_4(t_k-) > N_5(t_k-)} \log(a(N_4(t_k-) - N_5(t_k-))) \\
&- \mu_3 T - a \underbrace{\int_0^T (N_4(t) - N_5(t))\, dt}_{\sum_{t_k} (N_4(t_{k-1}+) - N_5(t_{k-1}+))(t_k - t_{k-1})}
\end{aligned}$$

where $H_j(T) = G_j(T) - G_j(t_N)$ and t_N is the last event arrival timestamps before T.

Therefore the gradients are straightforward:

$$\begin{aligned}
\frac{\partial L}{\partial \mu_i} &= \sum_{t_k, e_k = i} \frac{1}{\mu_i + \sum_{j=1}^{2} \alpha_{ij} g_j(t_k)} - T, \quad i = 1, 2 \\
\frac{\partial L}{\partial \alpha_{ij}} &= \sum_{t_k, e_k = i} \frac{g^j(t_k)}{\mu_i + \sum_{m=1}^{2} \alpha_{im} g_m(t_k)} - \sum_{t_k, e_k = i} H_j(t_k) - H_j(T), \quad i, j = 1, 2 \\
\frac{\partial L}{\partial \mu_3} &= \sum_{t_k, e_k = 4} \frac{1}{\mu_3} - T \\
\frac{\partial L}{\partial a} &= \sum_{t_k, e_k = 5} 1_{N_4(t_k-) > N_5(t_k-)} \frac{1}{a} - \sum_{t_k} (N_4(t_k-) - N_5(t_k-))(t_k - t_{k-1})
\end{aligned} \tag{4.C.4}$$

Some calculations

Given a complete path of X from $t = 0$ to t_n-, and the observable part of the data at time t_n, the probability of the color of n-th event being red is

$$\begin{aligned} p_\theta(Z_n = 1|Y_n, \mathcal{F}_{t_n-}) &= \frac{p_\theta(Z_n = 1, Y_n|\mathcal{F}_{t_n-})}{p_\theta(Y_n|\mathcal{F}_{t_n-})} \\ &= \frac{p_\theta(Z_n = 1, Y_n|\mathcal{F}_{t_n-})}{p_\theta(Z_n = 1, Y_n|\mathcal{F}_{t_n-}) + p_\theta(Z_n = 0, Y_n|\mathcal{F}_{t_n-})} \end{aligned} \tag{4.C.5}$$

For $e_n \in \{1, 2\}$,

$$\begin{aligned} \lambda_{e_n}(t_n) &= \lim_{dt \searrow 0} \frac{1}{dt} \mathbb{P}(N_{e_n}(t_n + dt) - N_{e_n}(t_n) = 1|\mathcal{F}_{t_n-}) \\ &= \mu_{e_n} + \alpha_{e_n,1} \sum_{t_k < t_n, r_k = 1} g_1(t_k) + \alpha_{e_n,2} \sum_{t_k < t_n, r_k = 2} g_2(t_k) \quad \text{is independent of } Z_{[1:n-1]} \end{aligned} \tag{4.C.6}$$

Therefore

$$p_\theta(Z_n = 0|Y_n = (t_n, 1), \mathcal{F}_{t_n-}) = \frac{\lambda_1(t_n)}{\mu_3 + \lambda_1(t_n)} \tag{4.C.7}$$

$p_\theta(Z_n = 0|Y_n = (t_n, 1), \mathcal{F}_{t_n-})$ is independent of $Z_{[1:n-1]}$ while $p_\theta(Z_n = 0|Y_n = (t_n, 0), \mathcal{F}_{t_n-})$ depends on the upcoming shot noise on node 2.

Sequential Monte Carlo Expectation-Maximization (SMCEM)

The Expectation-Maximization algorithm (Dempster et al., 1977) is an efficient iterative procedure for estimating the Maximum Likelihood Estimate (MLE) in the presence of missing or hidden data. Given a set of observable data, a set of latent data Z and a vector of unknown parameters θ, the likelihood function for the complete data is $L(\theta; Y, Z) = p_\theta(Y, Z \,|\, \theta)$. However, since Z is unobserved, the MLE $\hat{\theta}$ is obtained by maximizing the marginal likelihood of the observed data $p_\theta(Y \,|\, \theta)$. The EM algorithm is used when the marginal likelihood is difficult to compute. It accomplishes this task through an iterative process consisting of two key steps:

- **E-step**: $Q(\theta \,|\, \theta^{(k)}) = \mathbb{E}_{Z \sim p(\cdot|Y, \theta^{(k)})}[\log p(Y, Z \,|\, \theta)]$
- **M-step**: $\theta^{(k+1)} = \arg\max_\theta Q(\theta \,|\, \theta^{(k)})$

In our specific case, where the distribution $p_\theta(Z_{1:N}|Y_{1:N})$ lacks a closed-form expression, we need to approximate the expected value $Q(\theta, \theta^{(k)})$ using the Monte Carlo method. Without proving the convergence, we propose the following SMCEM algorithm for our model.

SMCEM

Z is a sequence of random variables with $Z_n \in \{0, 1\}$. Since events of N_5 are delayed replications of N_4. Given a realization $Y_{1:N} = (t, r)_{1:N}$, $Z_{1:N}$ should satisfy the following constraints:

$$\sum_{i=1}^{n} Z_i 1_{r_i=1} \geqslant \sum_{i=1}^{n} Z_i 1_{r_i=2}, \quad \forall n \in \{1, 2, \ldots, N\}$$

Let q be a probability of $Z_{1:N}$ which is defined as follows:

$$q(Z_n = 1 \,|\, r_n = 1, Y_{1:n-1}, Z_{1:n-1}) = \frac{1}{2}$$

$$q(Z_n = 1 \,|\, r_n = 2, Y_{1:n-1}, Z_{1:n-1}) = \begin{cases} 0 & \text{if } \sum_{i=1}^{n-1} Z_i 1_{r_i=1} \geqslant \sum_{i=1}^{n-1} Z_i 1_{r_i=2} \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

Then the SMCEM algorithm is as follows:

- **Monte Carlo** : for $m \in [1:M]$, generate $Z_{1:N}^{(m)} \sim q$.
- **E-step** :

$$\begin{aligned} Q(\theta, \theta^n) &= \mathbb{E}_{Z_{1:N} \sim p_{\theta^{(k)}}(z_{1:N}|y_{1:N})}\big[\log p_\theta(Y_{1:N}, Z_{1:N})\big] \\ &\approx \frac{1}{M} \sum_{m=1}^{M} \frac{p_{\theta^{(k)}}(Z_{1:N}^{(m)}|Y_{1:N})}{q(Z_{1:N}^{(m)})} \log p_\theta(Y_{1:N}, Z_{1:N}^{(m)}) \end{aligned}$$

- **M-step** : $\theta^{n+1} = \arg\max_\theta Q(\theta, \theta^n)$

Online SMCEM

- **Init**: Set initial parameters $\theta^{(0)}$, and calculate the conditional distribution of Z_1 given the initial event shape $p_{\theta^{(0)}}(Z_1 = 1|r_1 = 1) = \frac{\mu_3^{(0)}}{\mu_1^{(0)} + \mu_3^{(0)}}$, $p_{\theta^{(0)}}(Z_1 = 1|r_1 = 2) = 0$.
  For $m \in [1:M]$, draw $Z_1^m \sim P_{\theta^{(0)}}(Z_1|Y_1) = P_{\theta^{(0)}}(Z_1|r_1)$.
- **Iteration**: at n-th iteration (i.e., at n-th event)
  - **Monte Carlo** : for $m \in [1:M]$, draw $Z_n^m \sim P_{\theta^{(n-1)}}(Z_n|Y_{1:n}, Z_{1:n-1}^m)$.
  - **E-step** :

$$\begin{aligned} Q(\theta, \theta^n) &= \mathbb{E}_{Z_{1:n} \sim P_{\theta^{(n)}}(z_{1:n}|y_{1:n})}\big[\log P_\theta(Y_{1:n}, Z_{1:n})\big] \\ &= \sum_{z_{1:n}} P_{\theta^{(n)}}(Z_{1:n}|Y_{1:n}) \log P_\theta(Y_{1:n}, Z_{1:n}) \\ &= \sum_{Z_n} \sum_{Z_{1:n-1}} P_{\theta^n}(Z_n|Y_{1:n}, Z_{1:n-1}) P_{\theta^n}(Z_{1:n-1}|Y_{1:n}) \log P_\theta(Y_{1:n}, Z_{1:n}) \\ &= \sum_{Z_n} \frac{1}{M} \sum_{m=1}^{M} P_{\theta^n}(Z_n|Y_{1:n}, Z_{1:n-1}^m) \log P_\theta(Y_{1:n}, Z_{1:n}^m) \end{aligned}$$

  - **M-step** : $\theta^{n+1} = \arg\max_\theta Q(\theta, \theta^n)$

Remark 4.6. *In fact, we can reformulate the problem in a Markovian context.*

$$\begin{aligned} \boldsymbol{N} = (N_1, N_2, N_4, N_5) &\Leftrightarrow X = \{(t_n, e_n), n \in \mathbb{N}, e_n \in \{1, 2, 4, 5\}\} \\ &\Leftrightarrow \{(t_n, \underbrace{r_n, Z_n}_{e_n}), n \in \mathbb{N}, r_n \in \{1, 2\}, Z_n \in \{0, 1\}\} \\ &\Leftrightarrow \{(\underbrace{t_n, r_n}_{Y_n}, \underbrace{\sum_{k=1}^{n} (1_{e_k=4} - 1_{e_k=5})}_{=:\bar{Z}_n}), n \in \mathbb{N}, r_n \in \{1, 2\}, Z_n \in \mathbb{N}\} \end{aligned}$$

$(Y_n, \bar{Z}_n)$ is in fact Markovian:

- *once $(Y_{1:n}, \bar{Z}_{1:n})$ is given, $\lambda_i(t)$ is known for $i \in \{1, 2, 4, 5\}$ and $t \in [t_n, t_{n+1})$.*
- *From event n to event $n + 1$:*

- *if* r_{n+1} *(observable) is 1, then*
  - $\bar{Z}_{n+1} \leftarrow \bar{Z}_n + 1$ *with probability* $\dfrac{\mu_3}{\lambda_1(t_{n+1}) + \mu_3}$.
  - $\bar{Z}_{n+1} \leftarrow \bar{Z}_n$ *with probability* $\dfrac{\lambda_1(t_{n+1})}{\lambda_1(t_{n+1}) + \mu_3}$.
- *if* r_{n+1} *(observable) is 2, then*
  - $\bar{Z}_{n+1} \leftarrow \bar{Z}_n - 1$ *with probability* $\dfrac{\mu_3 \bar{Z}_n}{\lambda_2(t_{n+1}) + \mu_3 \bar{Z}_n}$ $(= 0$ *if* $\bar{Z}_n = 0)$.
  - $\bar{Z}_{n+1} \leftarrow \bar{Z}_n$ *with probability* $\dfrac{\lambda_2(t_{n+1})}{\lambda_2(t_{n+1}) + \mu_3 \bar{Z}_n}$.

CHAPTER 5

SUPERVISED LEARNING FOR CLASSIFICATION OF AGENTS

This chapter focuses on behaviors of agents in the market. We propose a supervised learning method to identify the agents, by using the Gated Recurrent Unit network. An input is a sequence of orders submitted by an agent and the output is the ID of this agent. The classification accuracy is very high, implying that the agents behave very differently from each other. We also analyze the importance of features and the embedding patterns of action types.

Contents

5.1 - Introduction

Nowadays, most modern financial markets use electronic limit order books (LOB), a double continuous auction system which tracks and records orders submitted by market participants (in this work, we also call them the "agents"). Modeling the limit order book is a key aspect to understand the microstructure of market. Existing researches on modeling limit order book can be roughly classified into two categories : the statistical model-based methods and data-driven machine learning methods.

Statistical model-based approaches usually require expert knowledge and certain assumptions about the system. A very rich literature exists on limit order book models, and some notable examples are mentioned here. One prominent model is the Zero-Intelligence (ZI) limit order book model, which

was proposed by Smith et al. (2003). This model describes the order flow using independent Poisson processes, under the assumption that orders are placed without any specific trading strategies. Despite its simplicity, the model was proven to be able to reproduce several important statistical properties (Daniels et al., 2003; Farmer et al., 2005). Since then, many more sophisticated works appeared. Remarkable work like Cont et al. (2010); Foucault et al. (2005) and Abergel and Jedidi (2013) gained recognition, along with the queue-reactive Markovian model (Huang et al., 2015) and the Hawkes models (Abergel and Jedidi, 2015; Large, 2007).

The use of data-driven machine learning approaches has become increasingly popular in finance due to advances in deep learning technology and the availability of large volumes of high-frequency trading data. Unlike traditional methods, these approaches do not necessarily require assumptions and can handle highly complex data. Most of the existing works focus on applying deep learning models on limit order book data for the purposes of price forecasting or price movement classification. For example, Sirignano (2019) proved the effectiveness of neural networks for limit order book modeling the distribution of the best bid/ask prices. Later, Sirignano and Cont (2019) tested a Deep LSTM network on 1000 US stocks, and exploited the universality of features across the stocks. Other different neural networks architectures were exploited to forecast the short-term price movement direction, including convolutional neural networks (DeepLOB) in Tashiro et al. (2019); Zhang et al. (2018, 2019), recurrent neural networks in Dixon (2018), LSTM and attention networks in Zhang et al. (2021) as well as self-attention transformer networks in Wallbridge (2020). Authors of Mäkinen et al. (2019) proposed a deep network that combined CNN, LSTM and attention mechanism to predict price jumps. In Maglaras et al. (2022), authors proposed a RNN architecture to estimate the distribution of time-to-fill for a limit order, while in Briola et al. (2020), different deep learning networks for forecasting price returns were studied in order to gain a comparative perspective. For more related works, on can refer to Doering et al. (2017); Jiang (2021); Kumbure et al. (2022); Nabipour et al. (2020); Ntakaris et al. (2018); Wang et al. (2018); Yan et al. (2020). We also recommend the book de Prado (2018) for more technical details.

The collective actions of agents on the limit order book determines the macroscopic evolution of the market. Therefore, to fully understand the dynamics of a market, it is crucial to comprehend the roles and strategies of individual agents. However, before delving into analyzing the behavior, strategies and impact of individual agents on the market, we first need to answer some fundamental questions. For instance, how do market participants differ from each other? Which features are the most important in identifying market participants ? In this work, we aim at answering these questions through a specific task which is to identify high-frequency agents.

Traditional statistical models for modeling the limit order book often rely on strong assumptions about the underlying system and may be limited in their ability to handle complex data. The aim of our work is to identify market participants by analyzing their actions in the limit order book. This data can be very complex as order actions can occur at different price levels, and the bid/ask prices, as well as the mid-price, can change frequently. This complexity makes it difficult to apply traditional mathematical models to the data. To overcome this challenge, we employ deep learning techniques to recognize market participants based on a sequence of consecutive orders they placed.

Deep learning is a powerful class of machine learning that is able to extract high-level features from the raw input by using multiple "deep" layers. However it usually requires access to large dataset. In this work, we focus on high-frequency agents who exhibit a significant level of activity, i.e., having a large number of orders placed during a given period. We apply a popular deep neural network, the Gated Recurrent Unit (GRU) (Cho et al., 2014), to learn a representation of these

agents' order sequences. Here a sequence of orders is defined as a list of consecutive orders placed by a market agent. To achieve the best quality of results, we conduct several scenarios with different lengths of the sequence, as well as different sets of features.

Outline. The rest of this chapter is organized as follows. In Section 5.2, we introduce the limit order book data used in this work and discuss the data preprocessing. Section 5.3 displays neural network architectures and feature engineering techniques used in our classification approach. Then in Section 5.4, we present the results of our study, including a comparison of classification accuracy across different models and scenarios. We also analyze the action types embedding patterns and the importance of features. To extend our analysis, in Section 5.5, we conduct additional experiments by increasing the granularity level of agent labels and based on these results we group the agents. Finally, we conclude our study in Section 5.6.

5.2 - Data Description

A Limit Order Book is a record of active orders, each order is represented by a separate row in the book. Each row includes the information about the order's characteristics and the current market context. The information about the order includes its arrival time, its side (sell or buy), the price at which it is placed, its size, its type (limit, cancellation or market) and the agent who placed this order. The current market context provides a snapshot of the bid and ask limit orders in the market.

In this study, we analyze the CAC40 future index limit order book (LOB) data obtained from the Euronext market. The data covers a period of 300 consecutive trading days, starting from January 6th, 2016 and ending on March 7th, 2017, and comprises data collected between 9:00 am and 5:00 pm for each day. The depth of the order book is up to 10 levels.

In the dataset, each order features two anonymized numeric IDs. The first ID represents the member who placed the order, and the second ID identifies the specific connection used. In this paper, we will refer to the first ID as "Member" and the second ID as "ITM". In high frequency trading, "ITM" stands for Interactive Trading Machine used by banks, hedge funds, and other financial institutions. "ITM"s are specialized computer systems which are designed to execute trades in a fast and efficient manner. It is important to note that ITMs are a division of Members, meaning an ITM belongs to a Member, while a Member can have multiple ITMs. In fact in our dataset, the number of ITMs used by each member can vary from one to more than one hundred.

5.2.1 Member selection

Out of our 300-day dataset, we have identified 170 active Members. Despite this, the majority of these members either have limited daily order volume or show only brief periods of activity. In this study, we only consider Members who have placed at least 6000 orders on more than 30 separate trading days. To determine this, we first identify the trading days where a Member placed at least 6000 orders, and then add that Member to our list of eligible Members if they meet the minimum threshold of 30 trading days. This selection process results in a pool of 28 highly active Members.

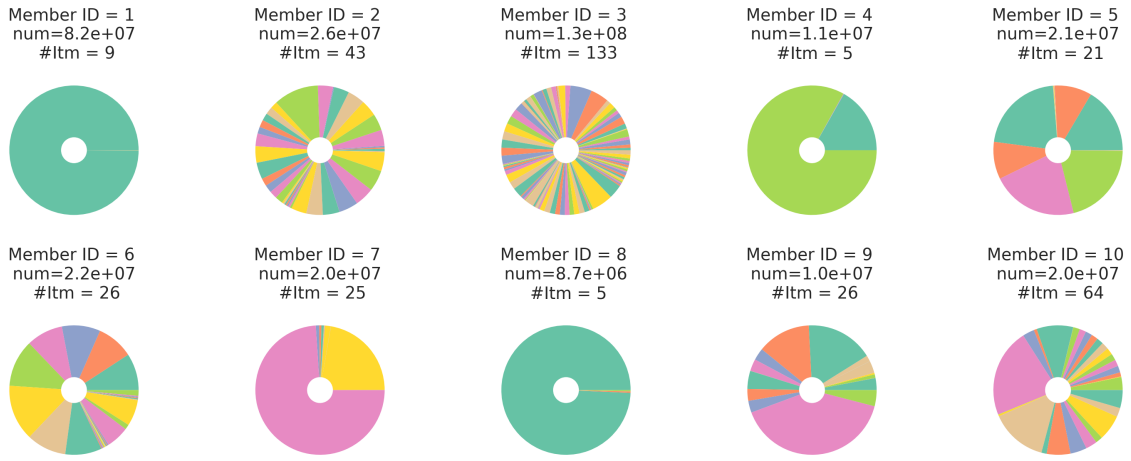


Figure 5.2.1 – Member–ITM. The figure displays the top 10 most active Member IDs and their associated ITM IDs. Each pie chart represents a Member, with each section of the pie representing a specific ITM. The sections are divided based on the proportion of orders associated with each ITM. The title of each pie chart consists of 3 parts : the Member ID, the total number of orders placed by that Member, and the number of ITMs represented within the chart.

5.2.2 Data normalization and extraction

Input

A single input, labeled as α, is a sequence of N orders placed by agent α. We denote a single input by $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N) \in \mathbb{R}^{N \times M}$ where $\forall i,\ \boldsymbol{x}_i \in \mathbb{R}^M$ is represented as a M-dimensional vector. In other words, each order is comprised of M features, which a subset of the feature collection presented in Table 5.3.1.

Data Normalization

Prior to extracting samples from the limit order book, we normalize the data as follows:

- time : $t_i - t_0$, where t_0 is the timestamps of the first order
- price : $\dfrac{P - \min(P)}{\max(P) - \min(P)} \in [0, 1]$, $\max(P)$ (resp. $\min(P)$) is the maximum (resp. minimum) of the prices in all limit order books from training days, including the best bid and the best ask prices $P_1^b(t-), P_1^a(t-)$ and $P_1^b(t+), P_1^a(t+)$,
- size : $\dfrac{V}{\max(V)} \in [0, 1]$, $\max(V)$ is the maximum of the volumes in all these limit order books from training, including the volumes of all limit orders $V_n^b(t-), V_n^a(t-)$ and $V_n^b(t+), V_n^a(t+)$.

Extraction of samples

After normalization, we extract 200,000 samples in total for training, validation, and testing days. The number of samples is divided into three parts with a 0.70:0.15:0.15 ratio for training, validation, and testing, respectively. Or to be more specific, for each agent α, we divide its available dates (i.e., the trading days where it has more than 6000 orders) into 3 parts : training dates $\mathbb{D}_{train}(\alpha)$, validation dates $\mathbb{D}_{val}(\alpha)$ and test dates $\mathbb{D}_{test}(\alpha)$, with the same proportion as before (0.7:0.15:0.15). The three sets of days are disjoint to prevent information leakage. The labels of these samples i.e.,

Feature	Value	#	Description
t_x (or t_i)	$\mathbb{R}^+$	1	time
$P_1^b(t_x-)$	$\mathbb{R}^+$	1	best bid price immediately before x
$P_1^a(t_x-)$	$\mathbb{R}^+$	1	best ask price immediately before x
$\{V_k^b(t_x-)\}_{k=1}^{10}$	$\mathbb{N}^+$	10	volume of bid side limit orders at kth level before x
$\{V_k^a(t_x-)\}_{k=1}^{10}$	$\mathbb{N}^+$	10	volume of ask side limit orders at kth level before x
P_x	$\mathbb{R}^+$	1	order price
V_x	$\mathbb{N}^+$	1	order size
A_x	$\{0,1\}$	1	indicates if the order is aggressive (i.e., a market order)
s_x	$\{0,1\}$	1	indicates if the order is on the ask side (1) or bid side (0)
$P_1^b(t_x+)$	$\mathbb{R}^+$	1	best bid price immediately after x
$P_1^a(t_x+)$	$\mathbb{R}^+$	1	best ask price immediately after x
$\{V_k^b(t_x+)\}_{k=1}^{10}$	$\mathbb{N}^+$	10	volume of bid side limit orders at level k after x
$\{V_k^a(t_x+)\}_{k=1}^{10}$	$\mathbb{N}^+$	10	volume of ask side limit orders at level k after x

Table 5.2.1 – Features for a single order $\boldsymbol{x}$ in the raw data, the ith order of a sequence of orders

their corresponding agents are uniformly distributed.

To extract a training sample of agent α, we randomly select a date D from $\mathbb{D}_{train}(\alpha)$, based on the the number of orders during a day over all days

$$\mathbb{P}(D) = \frac{\#\{\alpha\text{'s orders at day } D\}}{\sum_{D\in\mathbb{D}_{train}(\alpha)} \#\{\alpha\text{'s orders at day } D\}}, \text{ for } D \in \mathbb{D}_{train}(\alpha)$$

and randomly extract N consecutive orders of this agent on the selected day. The same process is repeated for the validation and test samples.

5.3 - Methodology

Deep learning is a powerful class of machine learning that is able to extract high-level features from the raw input by using multiple "deep" layers. In this work, we apply a popular deep neural network, the Gated Recurrent Unit (GRU), to learn a representation of these agents' order sequences. To achieve the best quality of results, we conduct several scenarios with different lengths of the sequence, as well as different sets of features.

5.3.1 Model architectures and Implementation

We detail our network architectures in this section, which depend on the selection of input features. In general, they comprise two main building blocks : Bidirectional Gated Recurrent Unit (GRU) layers and two dense layers.

Gated Recurrent Unit

Gated Recurrent Unit (GRU), introduced in 2014 by Cho et al. (2014), is a special type of Recurrent Neural Network architecture designed to solve the vanishing gradient problem of a standard RNN. It has two gates, the update gate and the reset gate. These gates are used to control the flow of information through the network. This design has led GRUs to be widely applied in various ares, including natural language processing, speech recognition, etc.

In this work, we use Bidirectional GRU (or simply BiGRU) to process the sequential data of orders. To recap, Bidirectional GRU is a variant of GRU architecture that allows information flow in two directions, i.e., both forward and backward in time. It is usually more powerful than a normal GRU as it captures contextual information from both past and future.

Dropout

Dropout (Srivastava et al., 2014) is a powerful regularization technique that can be used in neural networks to prevent overfitting. During training, it involves randomly setting a fraction of the neurons to zero, which can help prevent the network from relying too heavily on any one feature or a set of features.

To apply dropout effectively, it is important to set an appropriate dropout rate. In this work, we used the Python library Ray Tune (Liaw et al., 2018) to optimize the dropout rate. By doing so, we found that a dropout rate of 0.3 worked well for our neural network.

The problem in this work is to classify sequences of orders based on their associated Member. Therefore the output of the final dense layer is a 28-dimensional vector that goes through a SoftMax activation function to produce the final probability result, where the SoftMax activation function is defined by :

$$\boldsymbol{\sigma}(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_j^d e^{z_j}}, \text{ for } i = 1, 2, ..., d \text{ and } \boldsymbol{z} = (z_1, z_2, ..., z_d) \in \mathbb{R}^d$$

where d is the number of classes (in this work $d = 28$) and $\boldsymbol{z}$ is the input to softmax function. As $\boldsymbol{\sigma}(\boldsymbol{z})$ has each component in the range $(0, 1)$ and the components sum up to 1, it can be interpreted as a vector of probabilities. When the network is used to classify an input $\boldsymbol{x}$ sequence of orders, the prediction $\hat{y}$ will be determined by the label corresponding to the highest probability in the output distribution, i.e., $\hat{y} = \arg\max(\{\boldsymbol{\sigma}(\boldsymbol{z})_i\}_{i=1,2,...,28})$.

Implementation

The neural networks are mainly built using Keras (Chollet et al., 2018), an open-source Python library. We apply the neural networks architectures to 4 different input cases (see Table 5.3.1). The neural networks are equipped with cross-entropy loss and the ADAM (Adaptive Moment Estimation) (Kingma and Ba, 2015) optimizer is applied. The data is trained on training samples, the learning is stopped when the validation accuracy does not increase for 50 more epochs.

5.3.2 Feature engineering

In our raw data, an order consists of all the features which are displayed in Table 5.2.1. Now in order to enhance model accuracy, we will create the following new features :

- **Action type** : the action type is a categorical variable, transformed from the six rows at the bottom of Table 5.2.1, or the FMC features in Table 5.3.1.

The set of all possible action type categories is defined as

$$\begin{aligned}\mathcal{C} = &\{\text{Ask Limit at level k, Bid Limit at level k}, k \in [1:10]\} \\ &\cup \{\text{Ask Cancel at level k, Bid Cancel at level k}, k \in [1:10]\} \\ &\cup \{\text{Ask Market order, Bid Market order}\}\end{aligned} \tag{5.3.1}$$

- **Order flow** : the order flow is a new added feature which cannot be derived from Table 5.2.1. The order flow type C i.e., $F_c(t_{i-1} \to t_i)$ indicates the number of orders of type C in the entire market between two consecutive orders made by an agent. [1]

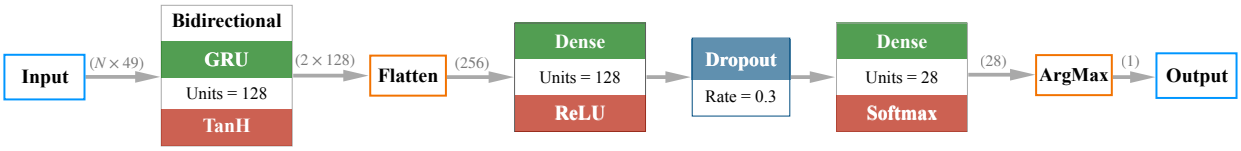
We will employ our model to evaluate three competitive sets of features, namely (1) HMC+FMC, (2) HMC+AT and (3) HMC+AT+OF as shown in Table 5.3.1. In addition, we will present a benchmark model which incorporates only three features which are considered as the most significant (from the results afterwards). The three sets of features share 25 common features, including 3 characteristics of the order and 22 market context indicators immediately before the execution of the order.

Class	Feature	#	Baseline	HMC+FMC	HMC+AT	HMC+AT+OF
HMC	t_x (or t_i)	1	✓	✓	✓	✓
	P_x (price)	1	✗	✓	✓	✓
	V_x (size)	1	✓	✓	✓	✓
	$P_1^b(t_x-)$	1	✗	✓	✓	✓
	$P_1^a(t_x-)$	1	✗	✓	✓	✓
	$\{V_k^b(t_x-)\}_{k=1}^{10}$	10	✗	✓	✓	✓
	$\{V_k^a(t_x-)\}_{k=1}^{10}$	10	✗	✓	✓	✓
FMC	A_x (aggressive ?)	1	✗	✓	✗	✗
	s_x (side)	1	✗	✓	✗	✗
	$P_1^b(t_x+)$	1	✗	✓	✗	✗
	$P_1^a(t_x+)$	1	✗	✓	✗	✗
	$\{V_k^b(t_x+)\}_{k=1}^{10}$	10	✗	✓	✗	✗
	$\{V_k^a(t_x+)\}_{k=1}^{10}$	10	✗	✓	✗	✗
AT	C_x (action type)	1	✓	✗	✓	✓
OF	$\{F_c(t_{i-1} \to t_i)\}_{c\in\mathcal{C}}$ (flow)	42	✗	✗	✗	✓

Table 5.3.1 – Features for a single order $\boldsymbol{x}$, the ith order of a sequence of orders

1. Normalization of the order flow : $\dfrac{\min\{F_c, 100\}}{100} \in [0, 1]$

HMC+FMC The first set of features (scenario 1) consist of the 25 basic features (HMC) and also the 24 supplementary features (FMC), as shown in Table 5.3.1. Therefore a single input of type 1 consists of in total N rows with each row having 49 features.



HMC+AT In the previous set of features, we supply the market context before and after an order (HMC+FMC), expecting the neural network to discover the full information of an order by itself. While in this set of features (scenario 2), instead of constructing the entire limit order book context after each order, we provide its action type. With the action type C_x of an order $\boldsymbol{x}$, we will be able to fully reconstruct the order book after the execution of order x.

Remark 5.1. *In the original order books, there are certain types of orders referred to as "modifications". These modifications indicate that the agent has either:*

(a) relocated some orders from one level to another (on the same side), or

(b) altered the quantity of a limit order.

In the first scenario (a), we divide the modification order into two separate orders: a cancellation order at the previous limit price and a limit order at the new price (with the same timestamp). In the second scenario (b), as the level of the order remains unchanged, it can be treated as either a cancel or limit order. If the size decreases, it is considered a cancel order, otherwise, it is considered as a limit order.

To summarize, an input of scenario 2 (HMC+AT) consists of 26 features in total, comprising 25 basic features (HMC) and 1 additional feature representing the action type (AT).

Since the action type C_x is a categorical variable, it will be encoded as a one-hot numeric array before being sent to GRU, i.e., to handle a category $C_x = i$, the one-hot encoding of C_x is represented as $\boldsymbol{C}_x = (0, 0, ..., \underbrace{1}_{ith \text{ element}}, ..., 0)$.

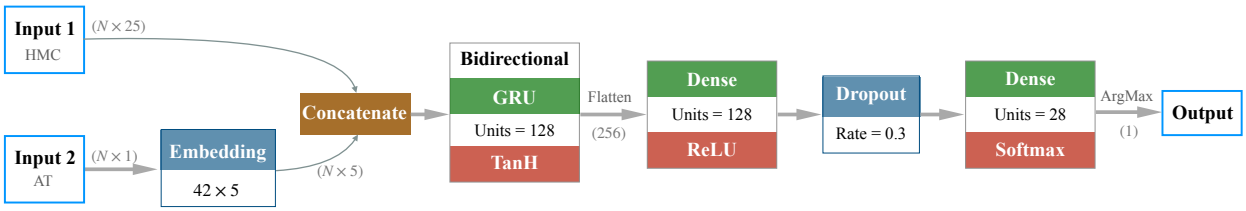


Figure 5.3.1 – NN .. input 1 is the 50 orders with 25 basic features, input_2 is the supplementary feature

The first layer is an embedding layer which converts one-hot vector $\boldsymbol{C}_x$ to a vector of dimension 5. To make it clearer, the operation of this layer is in fact $\boldsymbol{C}_x^{embed} = \boldsymbol{C}_x Q$, where Q is a matrix in $\mathbb{R}^{42\times 5}$.

We replace the scalar order type ($C_x \in [1:42]$) with a 5-dimensional embedding vector, which serves as input to a Bidirectional GRU layer with 30 features.

HMC+AT+OF If we look at the numerical results beforehand in Table 5.4.1, it appears that we have provided enough information of a particular agent based on the previous input it has received. However the model still lacks the understanding or the dynamics of the entire order book. Until now, our inputs only include the actions of the agent and the market context, but what we aim to achieve is to enable the model to also comprehend what happened in the market between two consecutive orders of this agent, or in other words, what drives the agent to take a specific action.

To enhance our model's ability, in addition to the 26 features from the second model (HMC+AT), we propose to add 42 additional order flow features (OF), represented by $\{\textit{Flow type } C, C \in \mathcal{C}\}$. We attempt to give a representation of the market dynamics by providing the flow of each action type. By doing so, we expect that the model can learn not only the aspects of an specific agent but also the interactions between the agent and the entire market. By adding these new features, the total number of features will be increased to 66.

5.4 - Numerical results

To establish a benchmark for comparison, we selected the LightGBM method as our reference model. LightGBM is a gradient boosting framework that uses tree-based learning algorithms introduced in Ke et al. (2017). It is often compared to deep learning algorithms because both are widely used in machine learning and have been shown to be effective in many applications. In this work, we compare LightGBM and BiGRU model, expecting that the deep learning model can capture more intricate patterns in the data. Table 5.4.1 presents a comparison of the accuracy of the two classification models when applied to sequences of different lengths.

Input Type	$N = 20$		$N = 50$		$N = 100$	
	BiGRU	LightGBM	BiGRU	LightGBM	BiGRU	LightGBM
Baseline	0.819	0.850	0.878	0.879	0.916	0.894
HMC+FMC	0.737	0.708	0.857	0.756	0.892	0.784
HMC+AT	0.866	0.861	0.912	0.889	0.943	0.900
HMC+AT+OF	**0.882**	0.846	**0.916**	0.885	**0.948**	0.905

Table 5.4.1 – Numerical results. This table displays the accuracy results of different models applied to the test data. (HMC = History Market Context, FMC = Future Market Context, AT = Action Type, ATF = Action Type Flow)

To learn more about the performance of our classification model, we define the accuracy of classifying samples for a given agent α as:

$$r_{acc}(\alpha) = \frac{|\{\hat{y}_n = \alpha | y_n = \alpha, n = 1, 2, ..., N\}|}{|\{y_n = \alpha, n = 1, 2, ..., N\}|}$$

Here, y_n represents the true label of the n-th sample and $\hat{y}_n$ represents the predicted label. $|\cdot|$ stands for the cardinality of a set. The accuracy rate r_{acc} for each agent is presented in Figure 5.4.1. We can see that the selected high-frequency agents exhibit considerable variability. Most agent demonstrate an accuracy rate of over 90%, with several agents (eg. 8,12 and 21) achieving nearly 100%.

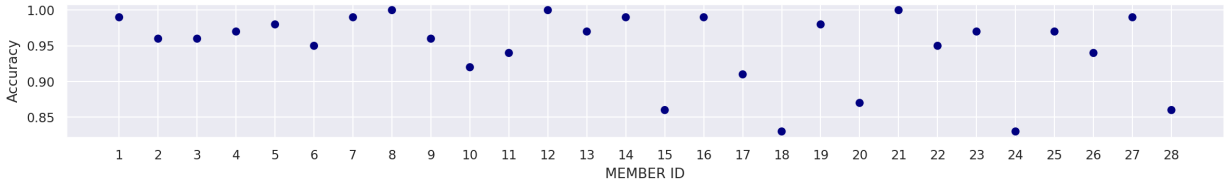


Figure 5.4.1 – Accuracy of classifying samples for each agent. The x-axis represents the member ID, while the y-axis represents the accuracy rate.

Embedding of action types

Inspired by the remarkable success of Word2Vec (Mikolov et al., 2013a,b), where the word embedding approach captures various degrees of semantic similarity between words (such as "man" - "king" = "woman" - "queen"), we are curious to explore if a similar approach can be applied to the embedding of action types in our work. Specially we are interested in examining if embedding action types would reveal any patterns or relationships between different categories of actions.

Figure 5.4.2 displays the results of action type embedding from the BiGRU N=10 input type (HMC+AT). An point in the figure represents the 2-dimensional Principal Component Analysis (PCA) projection of the 5-dimensional embedding vector of an action type. As shown in the figure, the embedding layer has effectively differentiated between the three order types - limit, cancellation and market - which demonstrates its efficacy in capturing the differences between this actions. Moreover, the figure reveals that cancellations at level 1 are closely aligned with market orders, as both types have the same impact on the market.

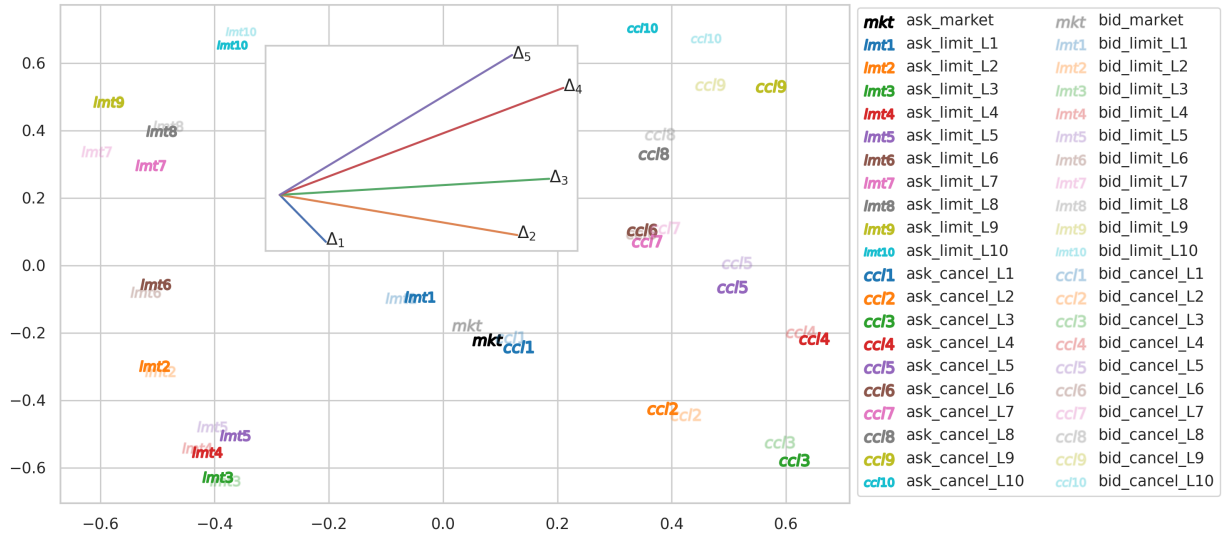


Figure 5.4.2 – Two-dimensional PCA projection of the 5-dimensional embedding vectors of action types. The inserted figure displays 5 lines, with each line $\Delta_i, i = 1, 2, 3, 4, 5$ indicating the vector from limit level i to cancellation level i.

Feature importance

In the following figures, we will show the feature importance matrices of LightGBM. The feature importance matrices offer a visual representation of the relative impact each feature has on the model's output. The importance of a feature is calculated by the number of times it is used to split the data across all trees. Features that are used more frequently are considered to be more important and have a greater impact on the model's prediction.

Figure 5.4.3 shows the feature importance of model (HMC+AT) when $N = 10$ while Figure 5.4.4 displays the feature importance of benchmark model. By analyzing both feature importance matrices, as well as the prediction accuracy results in Table 5.4.1, we observed that the order type and time (or frequency) are the most significant features.

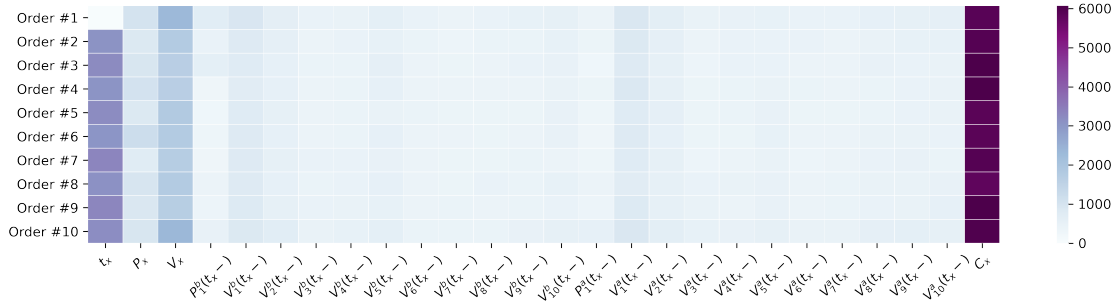


Figure 5.4.3 – Feature importance heatmap

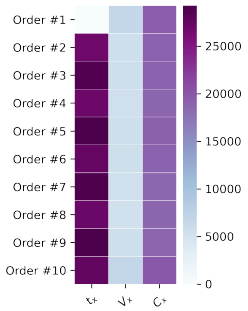


Figure 5.4.4 – Feature importance heatmap

5.5 - Extend experiments to ITMs

Now, instead of classifying Members, we apply the NN architecture to classify ITMs using the second set of features (HMC+AT). By doing so, we aim to get a deeper insight into the relationship between Members and ITMs.

ITM selection

We have identified more than 1000 active ITMs. However we only focus on the ITMs that correspond to the selected Members. The eligibility criteria for ITM selection is the same as for Member selection; an ITM is considered eligible if it has placed at least 6000 orders on more than 30 distinct

trading days. This filtering process leads to a list of 104 highly active ITMs, that belong to 21 Members.

Confusion matrix

The Confusion Matrix, represented by $\boldsymbol{C}$, is a useful tool for evaluating the accuracy of a classifier. Each entry $\boldsymbol{C}_{ij}$ represents the number of instances that were labeled as class i but predicted as class j. In this study, we utilize the normalized confusion matrix, which is obtained by dividing each entry $\boldsymbol{C}_{ij}$ by the sum of all entries in row i, i.e., $\boldsymbol{C}_{ij}/\sum_{k=1}^{104}\boldsymbol{C}_{ik}$.

Now if we examine the confusion matrix in Figure 5.5.1, it is clear that the misclassified samples are frequently confused with other ITMs that belong to the same Member. We assume that the neural networks struggle to differentiate between these ITMs due to their high degree of similarity. Based on this assumption, we can leverage the information from the confusion matrix to group the ITMs of a Member into distinct clusters based on their shared characteristics. Specifically, each cluster can represent a unique manner or mode of operation that the ITMs belonging to that group tend to exhibit. For instance, by extracting the sub-matrix of Member with ID=3, which is a 39×39 matrix, we can use the agglomerative hierarchical clustering to regroup the ITMs. The resulting groups of ITMs are considered to share similar behaviors, as demonstrated by dendrogram plots in Figures 5.5.2a and 5.5.2b. In the same way, Figure 5.5.3a and 5.5.3b show the clustering result for Member 6.

5.6 - Conclusion

We demonstrate in Section 5.4, that even the worst-case scenario has an average accuracy of over 70%, which means that the high-frequency agents are vastly different from one another. By analyzing the feature importance, we show that the most important features for the classification task are the time (or frequency), order size and action type of order.

We find that the best way to provide the action types to the neural networks is to categorize them and apply an embedding layer after one-hot encoding. By visualizing the projections of such embedded action types on $\mathbb{R}^2$, we surprisingly find that the neural networks have "understood" the relationship among these action types very well.

Based on these findings, we apply the best-performing model to an extended task. In Section 5.2, we introduce the concept of two IDs assigned to an agent, Member ID and ITM, where ITMs are a division of Member IDs. In contrast to the first classification task which classifies Member IDs, this additional task focuses on classifying ITMs. The confusion matrix shows clear diagonal blocks, with each block representing a Member ID. We then demonstrate that this is in fact a useful tool for clustering ITMs for a given Member ID.

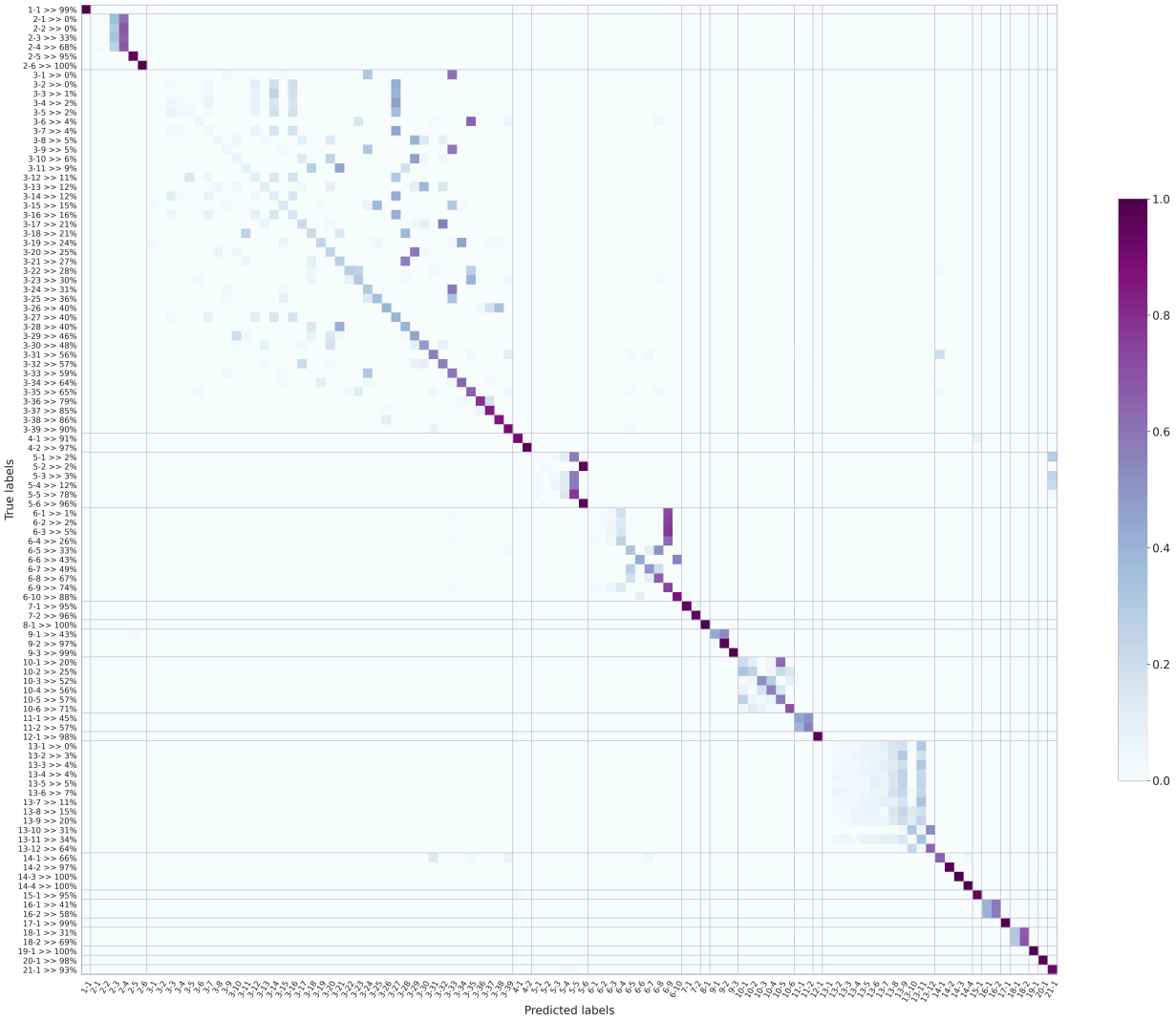
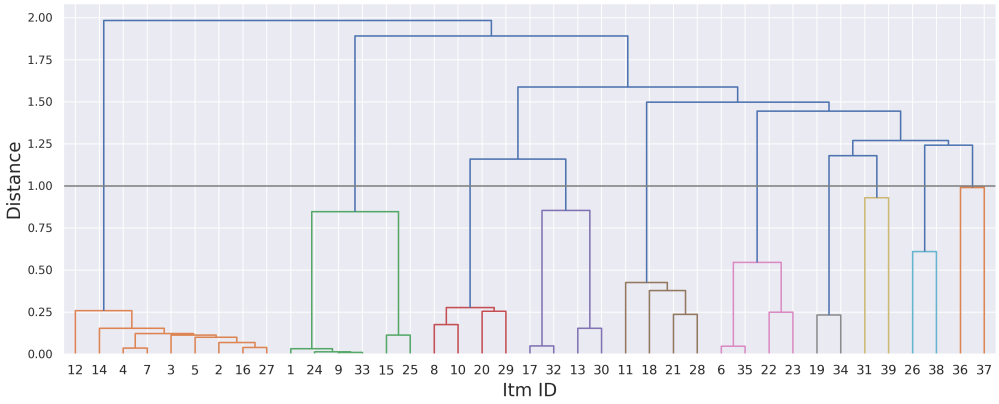
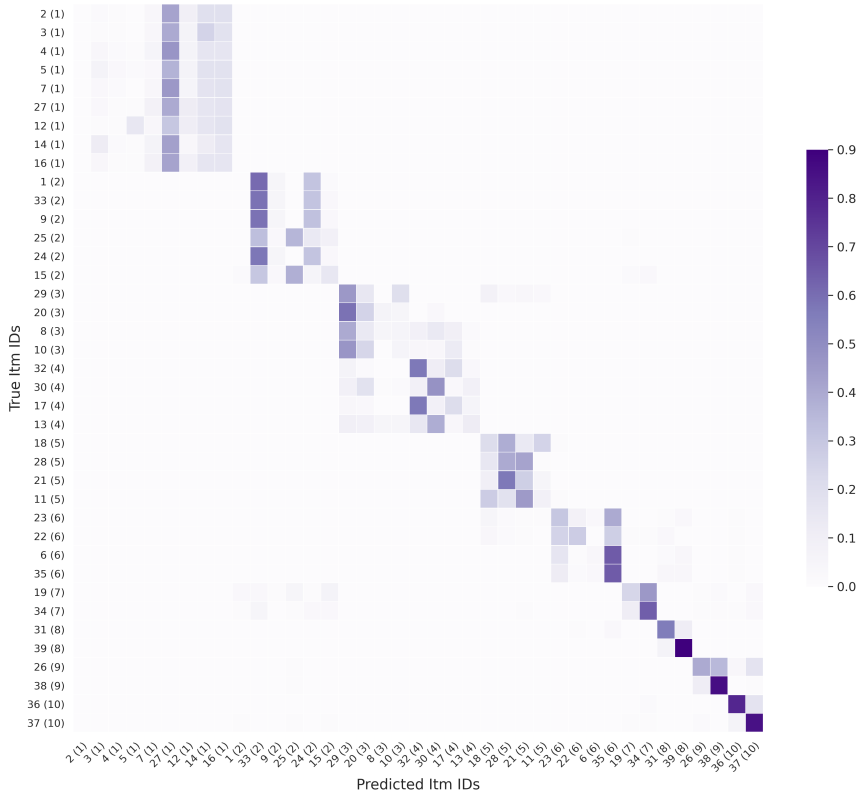


Figure 5.5.1 – Confusion matrix. The diagonal submatrix blocks represent the grouping of ITMs from the same Member. The labels along the x-axis (abscissa) are presented in the format "ITM - Member", while the labels along the y-axis (ordinate) are displayed in the format "ITM - Member >> Accuracy".

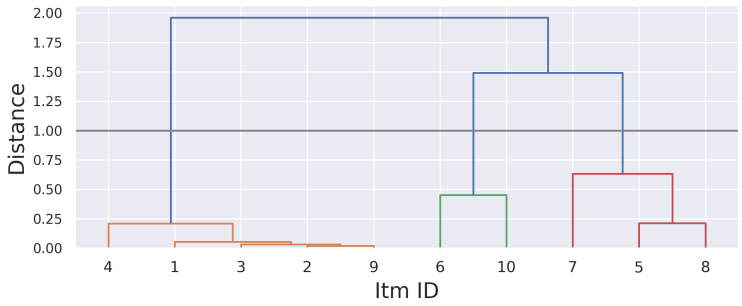


(a) Dendrogram plots, given a threshold 1, we can get 10 subgroups.

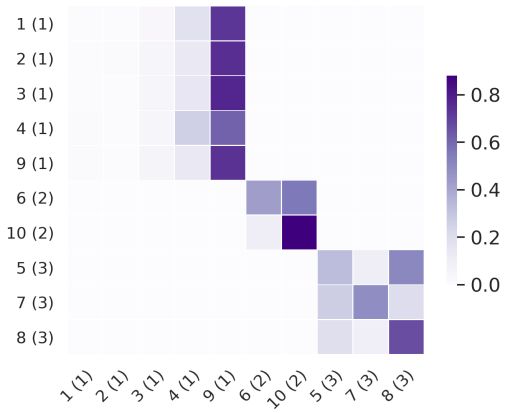


(b)

Figure 5.5.2 – Member ID = 3. The labels on both axes are presented in the format "ITM(cluster)", where "cluster" indicates the subgroup to which the ITM belongs.



(a) Dendrogram plots, given a threshold 1, we can get 3 subgroups.



(b)

Figure 5.5.3 – Member ID = 6. The labels on both axes are presented in the format "ITM(cluster)", where "cluster" indicates the subgroup to which the ITM belongs.

CHAPTER 6

SELF-SUPERVISED LEARNING FOR CLUSTERING AGENTS

From : Liquidity takers behavior representation through a contrastive learning approach (Ruan et al., 2023a)
R. Ruan, E. Bacry, J.-F. Muzy

Thanks to the access to the labeled orders on the CAC40 data from Euronext, we are able to analyze agents' behaviors in the market based on their placed orders. In this study, we construct a self-supervised learning model using triplet loss to effectively learn the representation of agent market orders. By acquiring this learned representation, various downstream tasks become feasible. In this work, we utilize the K-means clustering algorithm on the learned representation vectors of agent orders to identify distinct behavior types within each cluster.

Contents

6.1 - Introduction

Deep learning has achieved great success in recent years, mainly due to advances in machine learning algorithms and computer hardware. As a result, it has become an indispensable tool in a wide range of fields, both in research and in practical applications. Specifically, in finance, deep learning has been applied extensively to predict stock prices movements using limit order book data. This technique is particularly effective in handling complex data which statistical models often struggle to manage. Notable works in the recent literature include Sirignano and Cont (2019); Sirignano (2019); Zhang et al. (2021, 2019).

In particular, contrastive learning (CL) is a powerful technique in deep learning that has led to significant advances in representation learning. It has been widely applied, especially in vision domain, as demonstrated by the success of works such as Chen et al. (2020); Grill et al. (2020); He et al. (2020). In the domain of time-series analysis, CL has also shown great potential. For example, Contrastive Predictive Coding (CPC) of Oord et al. (2018) employed a latent space to capture historical information and predict future observations, and has demonstrated impressive results in speech recognition tasks. In healthcare, authors in Mohsenvand et al. (2020) applied CL on electroencephalogram data while Mehari and Strodthoff (2022) used it on electrocardiography data. In finance, CL has been used for stock trend prediction (Hou et al., 2021), and financial time series forecasting (Wu et al., 2020).

In financial markets, the collective actions of agents in the limit order book determines the macroscopic evolution of the market. Therefore, to fully understand the dynamics of a market, it is crucial to comprehend the roles and strategies of individual agents. However, due to the challenge of accessing confidential trading data, only a few studies have been conducted in the area of characterizing market participants. For example, Brogaard et al. (2010, 2014) have studied limit order book data with agents labeled as either High-frequency traders (HFTs) or Market makers (MMs). Hagströmer and Nordén (2013) has analyzed the different behaviors of HFTs and MMs. With access to agent identities, Kirilenko et al. (2017) classified the agents into HFTs, MMs, fundamental buyers, fundamental sellers and opportunistic traders, and studied their behavior before and after the flash crash of may 2010. A recent research in this area was conducted by Cont et al. (2023), where the authors analyzed limit order book data from the broker view and grouped the agents into four groups, for each they detailed descriptions of the properties. A study that deserves special attention in our research is the work Cartea et al. (2023). In their study, the authors presented statistical models designed to predict the behavior of trading algorithms using data from Euronext Amsterdam. By extracting the coefficients from their prediction model, they identified three distinct categories of trading algorithms prevalent in the market: directional trading, opportunistic trading, and market making.

In this chapter, our objective is to analyze and characterize the different behaviors of the agents. More specifically, we will study successions of any fifty consecutive market orders placed by any agent. Each order is defined by eight features (see Section 6.3), resulting in a sample matrix of size 50×8. The inner structure of such a sample can be complex and challenging to represent using classical methods. To address this challenge, we aim to learn a representation (i.e., embeddings) that can effectively embed each such sample into a lower-dimensional vector space. We propose to use a self-supervised contrastive learning approach using a triplet loss (Schroff et al., 2015). For each "anchor" sample from an agent two other samples (not overlapping in time) are chosen :

- one (positive) sample from the same agent

- one (negative) sample from another agent.

The pretext task, from which the embeddings are learned, consists in trying to identify the positive sample (through the use of the triplet loss). All samples are taken over a two hours period of time during the same day, so that the positive sample and the reference sample, corresponding from the same agent and being close in time, can be considered hopefully as corresponding to the "same" structure/strategy.

The learned embeddings can be utilized for various downstream tasks, such as clustering and classification. In this work, we will apply the K-means clustering algorithm to the learned embeddings, by doing so, we aim to reveal the different strategies employed by agents and the development of their strategies over time. To the best of our knowledge, we are the first to propose a contrastive learning method on limit order book representation.

Outline. This chapter is organized as follows. In Section 6.2, we recall some important concepts and terminology, including limit order book, liquidity makers and takers, and contrastive learning with triplet loss. We will also properly formulate the main problem to be addressed in this chapter. In Section 6.3, we describe the data used as well as the preprocessing steps. Section 6.4 presents the neural network architectures, implementation information and evaluation metrics. We demonstrate the importance of some features. In Section 6.5, we apply the K-means clustering algorithm, a downstream task, to the learned embeddings. We analyze the properties of each cluster and the clustering results of each agent. Finally, concluding remarks and discussions are provided in Section 6.6.

6.2 - Preliminaries

In order to provide a comprehensive understanding of the concepts and terminology used in this chapter, we will begin by briefly reviewing several important concepts. These include concepts in financial markets background, such as limit order books, liquidity takers, as well as a loss function for contrastive learning, namely the triplet loss. At the end of this section, we will proceed to formulate the main problem to be addressed in this study.

6.2.1 Limit order book

A **limit order book** (LOB) is an auction mechanism used in financial markets to record the buy or sell orders placed by traders. These orders can be categorized into three major types: limit orders, cancellation orders, and market orders.

A **market order** is an order to buy or sell a stock at the market's current best available price, which typically ensures an immediate execution. Conversely, a limit order is a buy or sell order at a specific price, which cannot be executed immediately. This is because the current market quotes do not match the trader's desired target price. In this case, the limit order will join the queue in the limit order book and wait until it can be executed at the desired price or a better one, unless it has been canceled. The action to cancel a limit order corresponds to a cancellation order, which removes an unfilled order from the queue. We refer the interested readers to the reference Gould et al. (2013) for a very nice review of limit order book concepts.

Furthermore, it is important to provide definitions for **aggressive trades** and **passive trades**. When agent α places a market order, it effectively can be seen as a match of two orders. They are

respectively an existing limit order at price p in the queue placed by agent β, and a marketable limit order placed by agent α that matches this price p. This market order can be seen as both an aggressive trade for agent α and a passive trade for agent β. It is worth noting that these terminologies may differ from other definitions that one can find elsewhere.

6.2.2 Liquidity takers *vs.* Liquidity providers

In financial markets, participants can be broadly classified into two categories : liquidity providers and liquidity takers. Liquidity providers, also known as market markers, are the agents who place limit orders on both sides of the market (buy and sell) and attempt to earn the bid-ask spread. Conversely, **liquidity takers**, typically traders and investors, seek to earn profits from the price movement of asset or use the price movement as a hedge to the other positions in their portfolio. In traditional markets, market makers are usually designated by the market while in modern markets, anyone can be a market maker. In fact the distinction between liquidity providers and takers is not clear-cut.

In this chapter, we will focus on the the behavior of liquidity takers through the analysis of their aggressive trades in the LOB. Later in this chapter, we will demonstrate that some highly active liquidity takers are also significant liquidity providers. For instance, members 11, 12 and 24 in Fig. 6.3.1 can be considered as such agents.

6.2.3 Self-supervised learning with Triplet loss

Self-supervised learning (SSL) is a machine learning approach, which processes unlabeled data to obtain useful representations that are helpful for various downstream tasks. Among self-supervised methods, contrastive learning is a very popular technique, notably used for computer vision tasks (for instance SimCLP (Chen et al., 2020), BYOL (Grill et al., 2020), MoCo (He et al., 2020), Barlow twins (Zbontar et al., 2021)). Its aim is to learn a representation function that embeds similar inputs close together and dissimilar inputs far apart. Over the years, the loss functions used in contrastive learning have evolved from a simple comparison between one positive and one negative sample (Chopra et al., 2005) to multiple positive and negative samples (Gutmann and Hyvärinen, 2010; Oord et al., 2018; Sohn, 2016).

In this work, we apply our deep neural networks, which are equipped with the triplet loss, to learn a representation function for limit order book data. The Triplet Loss was first introduced by Schroff et al. (2015), where it was used for face recognition of individuals under varying poses and angles. Since then, it has become a widely used loss function for supervised similarity tasks. As illustrated by Fig.6.2.1, the fundamental idea behind the Triplet Loss is to learn a representation function $f(\cdot)$ that brings inputs that match (referred to as positive inputs) closer to the reference input (referred to as the anchor) and pushes away inputs that do not match (referred to as negative inputs). The triplet loss function can be defined as following :

$$\mathcal{L}_{triplet} = \sum_{i=1}^{N} \max\left(\|f(X_i^a) - f(X_i^p)\|_2^2 - \|f(X_i^a) - f(X_i^n)\|_2^2 + \gamma, 0\right) \qquad (\mathcal{L}_1)$$

where

- $f : \mathcal{X} \to \mathbb{R}^d$ is a representation function that embeds an input to a d-dimensional Euclidean space $\mathbb{R}^d$.
- γ is a margin between positive and negative pairs, the margin value is added to push negative samples far away.

- X^a indicates **Anchor** sample, X^p indicates **Positive** sample, X^n indicates **Negative** sample.

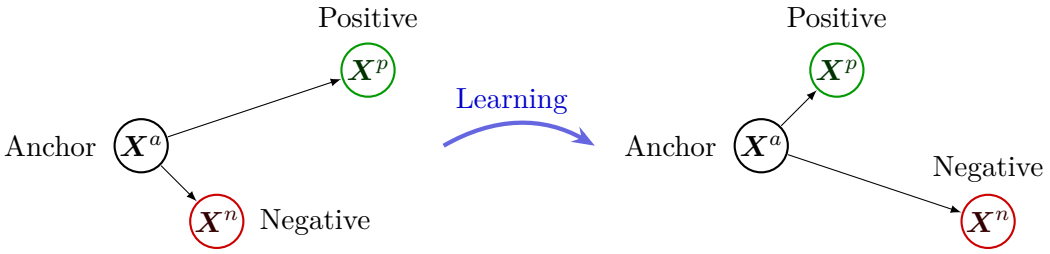


Figure 6.2.1 – Triplet Loss illustration. The Triplet Loss minimizes the distance between the anchor X^a and the positive X^p, and maximizes the distance between the Anchor X^a and the negative X^n.

6.2.4 Problem formulation

The objective of this work is to develop a robust method for representing a sequence of consecutive market orders sent by the same agent. To this end, we introduce a novel approach that employs a deep neural network with an LSTM architecture, equipped with a triplet loss function. Fig. 6.2.2 gives an illustration of an example framework (when $d = 2$).

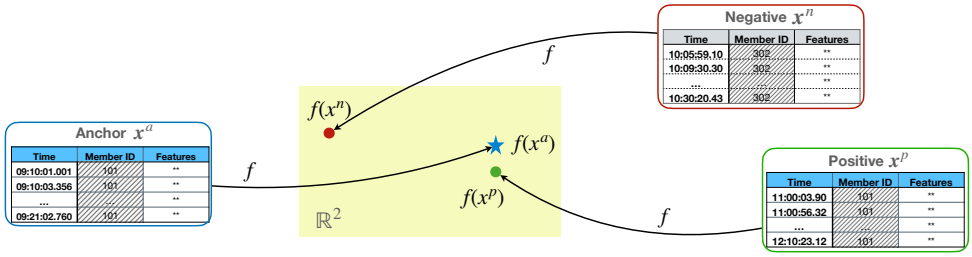


Figure 6.2.2 – Illustration of model learning. X is a sequence of consecutive market orders and $f(X)$ is its vector representation in $\mathbb{R}^2$. The triplet loss minimizes the distance between the samples from the same agent $\|f(X^p) - f(X^a)\|_2$ and maximize the distance between the samples from different agents $\|f(X^n) - f(X^a)\|_2$.

Since a sequence of orders is highly structural, determining the similarity between two sequences of orders can be very challenging. However the $\mathbb{R}^d$ space is widely recognized and offers a straightforward measure of distance between two points. As a result, the representation function f establishes a connection between the intricate order book space and a more comprehensible $\mathbb{R}^d$ space.

Once a comparison between two sequences of orders becomes possible, one can apply various downstream tasks. In this work, our focus lies in grouping these sequences of orders from different agents, represented by their images in $\mathbb{R}^d$, to several clusters. Through this clustering process, we expect to uncover the trading behavior and strategy of these agents.

6.3 - Data Description

In this present work, we analyze the limit order book (LOB) of the front month [1] CAC40 index futures contracts. The data was obtained from the Euronext market and spans a period of 300

1. The term "front month" refers to the nearest expiration date in futures trading.

consecutive trading days, from January 6th, 2016 to March 7th, 2017, between 9:00 am and 5:00 pm each day. Let us again mention that this task focuses only on market orders in the LOB rather than all types of orders.

We present a network that utilizes the Triplet Loss ($\mathcal{L}_1$) and takes consecutive market orders of an agent as inputs. Through training this network, our goal is to obtain a robust representation function that maps order book inputs to a lower-dimensional vector space. Similarity between two order book inputs is determined based on whether they belong to the same agent or not.

Agents selection

Out of our 300-day dataset, we have identified 170 active Members. Despite this, the majority of these members either have limited daily order volume or show only brief periods of activity. In this study, we only consider Members who have placed at least 200 market orders each day on more than 45 separate trading days. This selection process results in a pool of 30 highly "market order" active Members. Remarkably, each of these selected agents has placed no less than 15,000 market orders during the designated period.

Once more, we only consider the aggressive trades (market orders) executed by an agent during a period and do not include the passive trades, which are executed by market orders placed by another agent that fully or partially match the limit orders of this agent. However, it's important to note that passive trades can have a significant impact on an agent's behavior, and we plan to address this issue in our future work.

In order to have a deep insight into the 30 members that we selected and measure how much these 30 members are weighted in the market, we have conducted the following statistics.

- Let us define the "actions at L1" as all the orders (limit, cancellation and market) which are executed at the best bid or best ask price levels. The previous 30 selected members (which place the most market orders) are in fact among the top 40 most active agents at L1, with the top 20 most active agents included in this group. In simpler terms, these 30 selected members can be considered the most influential agents at L1.
- We also provide the ratio of the number of passive trades to the number of aggressive trades for each agent in Fig 6.3.1, in order to gain a better understanding on the visible of these agents. A high passive-aggressive ratio indicates a more market-marker-like agent, who provides more liquidity than takes liquidity from the market.

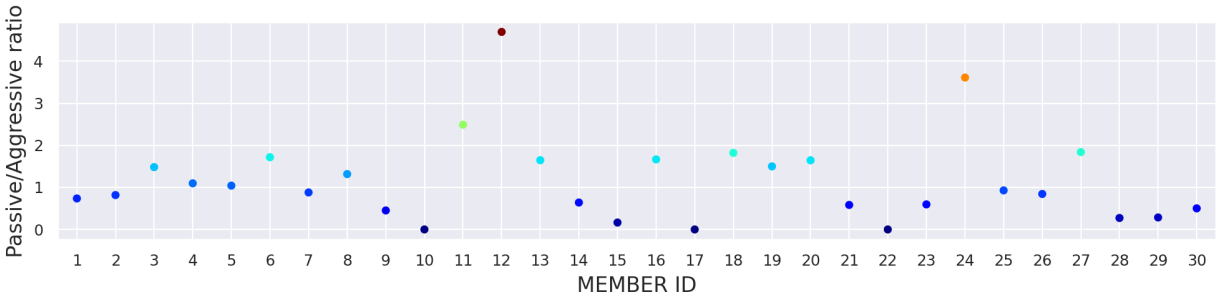


Figure 6.3.1 – The ratio of passive trades to aggressive trades for each Member

These statistics demonstrate that by focusing on those who place significantly aggressive trades, we

have effectively taken into account the majority of the most important active market participants.

Order features and Input data

In this work, we have chosen to describe a market order $\boldsymbol{x}$ by the following features,

- t (timestamps): the timestamp which records the time point when an order is executed
- q_T (Quantity): the amount of stocks traded (this value is always equal to or less than the size proposed by the trader)
- s (Side): whether an order is a buy or sell order
- M (Limit to trade modification): the specific type of action, $M = 1$ represents the modification of an existing limit order to make it aggressive, while $M = 0$ signifies the placement of an aggressive order that is immediately executed
- $P_1^b(t-)$: the best bid price immediately before the execution of this order
- $P_1^a(t-)$: the best ask price immediately before the execution of this order
- $Q_1^b(t-)$: the volume of limit orders at the best bid (best bid queue size)
- $Q_1^a(t-)$: the volume of limit orders at the best ask (best ask queue size)

It results that an order is represented as an 8-dimensional vector. Each input X is a sequence of 50 consecutive market orders executed by one agent α, $X = (\boldsymbol{x}_i)_{i=1,2,\dots,50}$, corresponding to a matrix in $\mathbb{R}^{50\times 8}$. The set of inputs labeled by α is denoted by $\mathcal{Y}_\alpha$. In the Triplet Loss approach, a single input is composed of X^a, X^p, X^n, where X^a and X^p (note that $X^a \neq X^p$) come from the same agent α, while X^n is sourced from a different agent β. In other words, X^a and X^p belong to the set $\mathcal{Y}_\alpha$, while X^n belongs to another set $\mathcal{Y}_\beta$, with $\beta \neq \alpha$. The triplets are constructed locally in time, the positive sample X^p and the negative sample X^n are required to be "temporally close" to the anchor sample. In this work, two samples are considered "temporally close" if the time interval between their first order's timestamps is less than 2 hours.

The use of a local model in this work is motivated by the dynamic nature of agent behavior. As the goal is to learn the representation of sequences of orders of an agent, it is expected that an agent's strategy may change over time. In such a scenario, it would not be appropriate to force inputs that are far apart in time to match, even though they are from the same agent. By utilizing a local model, the contrastive learning approach is leveraged to better capture the dynamic nature of agent behavior.

6.4 - Implementation details and numerical results

In this study, we employ Long Short-Term Memory (or simply LSTM) to process the sequences of market orders. LSTM is a variation of RNN which was designed to address the problem of vanishing gradients in standard RNNs (Hochreiter and Schmidhuber, 1997). Compared to another variant Gated recurrent unit (GRU), LSTM has a more complex structure. It includes memory cells, input gates, forget gates and output gates. Specifically, in this work, we applied stacked LSTMs, which are well-known for their ability to handle more complex models and deliver improved performance compared to the simple LSTM architecture (Sutskever et al., 2014).

6.4.1 Inputs and Hyperparameters

We conducted tests on multiple sets of input sample features, and we present three of these sets in Table 6.4.1. In the next subsection, we will introduce an evaluation metric and demonstrate that

the best set of features is the (Basic+M+QS). Additionally, we performed tests with an extended set of features, including the queue sizes at level 2 and 3 in addition to the eight features, however we found that they do not exert a significant impact on the current task. As a result, we conclude that these eight features listed in Table 6.4.1 are sufficient for our task.

Features	Basic	Basic+M	Basic+M+QS
Time (interevent time)	✓	✓	✓
Quantity	✓	✓	✓
Side (buy or sell)	✓	✓	✓
Limit to trade modification	✗	✓	✓
Best bid price	✓	✓	✓
Best ask price	✓	✓	✓
Best bid qty	✗	✗	✓
Best ask qty	✗	✗	✓

Table 6.4.1 – 3 types of input : Basic, Basic+M, Basic+M+QS. "M" stands for limit to trade modification and "QS" stands for the best level queue sizes.

The LSTM network used in this work has a stacked architecture with two hidden layers. The first layer consists of 100 units, while the second layer has 40 units. The encoded representation of the input sequence is obtained from the last output of the second layer, i.e., the dimension of the embedding space is $d = 40$ (We tried also other smaller output dimensions but they did not perform as well). The margin in triplet loss γ was set to 0.5. See Fig 6.4.1 for a model architecture illustration.

$$
\begin{aligned}
\textbf{Input} &: \text{triplet } (X^a, X^p, X^n) \\
\textbf{Encoder} &: f = f^2 \circ f^1 \text{ with } \begin{cases} f^1(\cdot) = LSTM(p \to 100) \\ f^2(\cdot) = LSTM(100 \to 40) \end{cases} \\
\textbf{Output} &: (f(X^a), f(X^p), f(X^n)) \\
\textbf{Loss} &: \max\left(\|f(X^a) - f(X^p)\|_2^2 - \|f(X^a) - f(X^n)\|_2^2 + \gamma, 0\right)
\end{aligned}
\tag{NN1}
$$

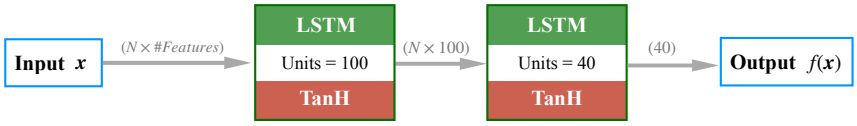


Figure 6.4.1 – Encoding model architecture schema for one sample.

The PyTorch library (Paszke et al., 2019) was primarily used to implement the neural networks. To train the networks, we used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.002 and a batch size of 64. The training process was conducted on a single NVIDIA GPU Tesla

P4. The training process was stopped after 500 epochs. And early-stopping was not implemented due to the the effectiveness of the triplet loss in preventing overfitting.

6.4.2 Numerical results

To prevent any data leakage, the 300 consecutive trading days are divided into two distinct sets, namely training days and test days. The numbers of training days and of test days follow a ratio of 4:1. Therefore the training inputs $\mathcal{T}_0$ and the test inputs $\mathcal{T}_1$ are extracted from 240 training days and the remaining 60 test days respectively. We introduce an evaluation metric for the test data $\mathcal{T}_1$, called the failure rate. This metric is defined by the following formula:

$$r = \frac{|\ \{i \in \{1, 2, ..., N\} \text{ such that } \|f(X_i^a) - f(X_i^n)\| < \|f(X_i^a) - f(X_i^p)\|\}\ |}{N}$$

where $N = |\mathcal{T}_1|$, and the test data $\mathcal{T}_1$ is defined as $\mathcal{T}_1 = \big\{(X_i^a, X_i^p, X_i^n), i = 1, 2, ..., N\big\}$. More precisely, when considering a particular agent α, the failure rate can be expressed as follows :

$$r_\alpha = \frac{|\ \{i \in \{1, 2, ..., N\} \text{ such that } X_i^a \in \mathcal{Y}_\alpha \text{ and } \|f(X_i^a) - f(X_i^n)\| < \|f(X_i^a) - f(X_i^p)\|\}\ |}{|\ \{i \in \{1, 2, ..., N\} \text{ such that } X_i^a \in \mathcal{Y}_\alpha\}\ |}$$

Here $|\ \cdot\ |$ stands for the cardinality of a set. The quantity $(1 - r_\alpha)$ is the proportion of triplets where the positive and negative sample are correctly distinguished.

Table 6.4.2 gives the unconditional failure rates of the 3 feature sets. It reveals that the action type and the best bid/ask price level queue sizes play a crucial role in this task. The evaluation comparison, conditional on agents, for these feature sets is illustrated in Figure 6.4.2.

Features	Basic	Basic+M	Basic+M+QS
Failure rate (r)	8.03%	6.72%	5.32%

Table 6.4.2 – Evaluation results for the 3 types of input : Basic, Basic+M, Basic+M+QS.

Figure 6.4.2 – Failure rate for each agent in different scenarios.

6.5 - Downstream task: Clustering

So far, we have acquired a learned representation function for market order sequences. In this section, we will test this representation in a downstream task that consists in performing a clustering of agent behavior.

Cluster analysis is the task of grouping or segmenting a collection of objects into subsets or "clusters", such that the objects within the same cluster are more similar to each other than those assigned to different clusters (see 14.3 in Hastie et al. (2009)). As detailed below, we will apply K-means cluster analysis to the encoded orderbook samples. To ensure a comprehensive analysis, we extract over 3000 samples for each agent from the market orders. Trough the neural network equipped with triplet loss, we obtain an encoder function, f, which can map a sequence of market orders to a lower dimensional vector that is interpretable. With the clustering of these lower dimensional vectors, we hope to uncover patterns that are not possible to discover through traditional statistical methods. More precisely, we aim to group the sequences of market orders by different agents into several subsets and each subset will be considered as a trading strategy. Let us note that an agent can belong to multiple clusters and a cluster may encompass samples from different agents.

6.5.1 K-means clustering

K-means clustering (Lloyd, 1982; MacQueen, 1967) is one of the most popular clustering methods. Given a set of observations $(\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n) \in \mathbb{R}^{n \times d}$, K-means clustering seeks to minimize the within-cluster sum of squared deviations by assigning each observation $\boldsymbol{y}_j$ to its nearest cluster center. To formulate the task mathematically, K-means algorithm assigns these observations to k^* clusters $\{Y_1, Y_2, ..., Y_{k*}\}$ by solving the following optimization problem. The cluster centers, denoted by μ_i, are updated iteratively until convergence.

$$\min \sum_{i=1}^{k^*} \sum_{\boldsymbol{y} \in Y_i} \|\boldsymbol{y} - \mu_i\|^2$$

In practice, in order to apply K-means, one must select the number of clusters k^*. In this work, we aim to apply K-means clustering to group the encoded orderbook samples into subsets of similar strategies. In order to provide a comprehensive view of strategies, it is desirable to have a relatively small number of clusters compared to the total number of agents. A variety of methods have been proposed in the literature to determine the number of clusters k^*, such as the "Elbow" method, the gap statistic, the Silhouette method, etc. In this work, we employ the "Elbow" method to determine the optimal number of K-means clusters, which is set to 7. Figure 6.5.1 displays the K-means clustering result.

One can remark that several Members belong to only one cluster, such as members 1,11,12,17 and 22. It is not surprising as these members have consistent and especially low failure rate (as shown in Figure 6.4.2), showing that they are rather distinctive. On the other hand, many agents belong to several clusters which may be interpreted by the fact that their strategy is changing over time.

6.5.2 Characterizing clusters by indicators

In order to have a deeper understanding of these clusters and how they differ from one another, we will evaluate them using a set of indicators. To start, we present each indicator and plot the evaluation of input data for each cluster based on this indicator. These box plots show the quantile

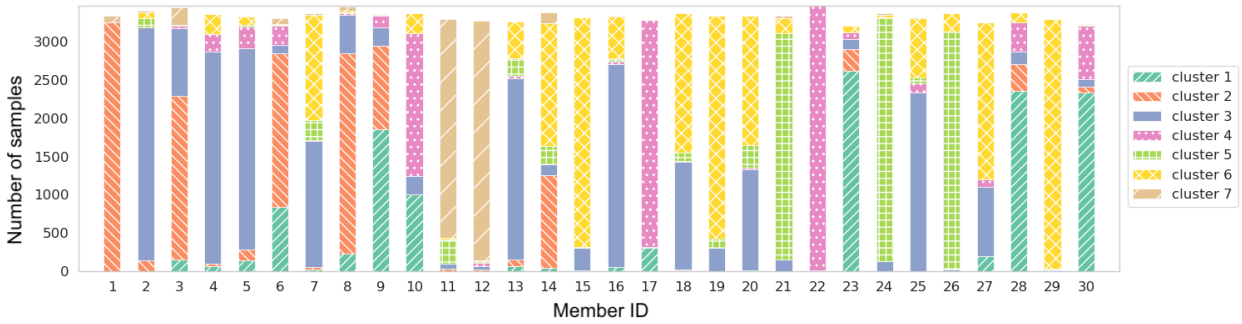


Figure 6.5.1 – K-means clustering results. Each agent is represented by a vertical bar, which may consist of one or multiple segments. Each segment corresponds to the agent's samples assigned to a specific cluster.

range (25th percentile, median and 75th percentile) for the inputs of each cluster.

Frequency: If δt stands for the average inter-event time (calculated as $\frac{1}{49}(t_{50}-t_1)$), the frequency indicator $\dfrac{60}{\delta t}$ represents the average number of trades per minute. The higher the value, the more frequently market orders are being placed by the agent.

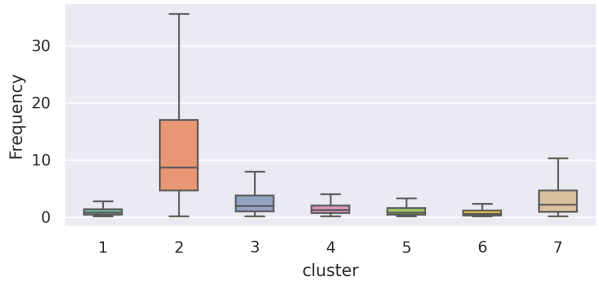


Figure 6.5.2 – Box plot of trade frequency data for samples within each cluster. The median is represented by the middle line. The box encompasses the lower (25%) and upper (75%) quartiles. The whiskers, extending from the box, indicate the range from the minimum to the lower quartile and from the upper quartile to the maximum values.

Size: Let $\hat{q}_i$ indicate the size of ith order, the average **order size** is denoted by $\dfrac{1}{50}\sum_{i=1}^{50}\hat{q}_i$. Additionally, we introduce another term called the average **trade size** $\dfrac{1}{50}\sum_{i=1}^{50}q_i$ where q_i is the filled quantity of ith market order. It is worth noting that $\hat{q}_i$ and q_i are not always the same, $\hat{q}_i$ represents the intended trade size, while q_i represents the actual executed trade size, therefore $q_i \leqslant \hat{q}_i$. With the two terms, we will be able to construct another indicator **fill rate**, which is calculated as $\sum_{i=1}^{50} q_i/(\sum_{i=1}^{50}\hat{q}_i)$ (always $\leqslant 1$).

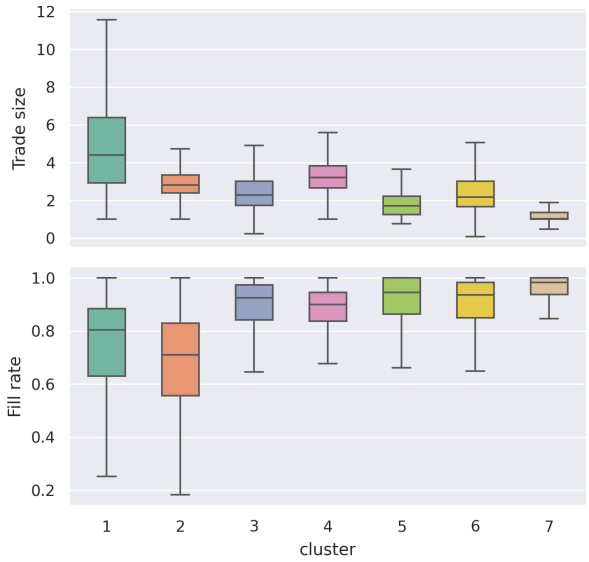


Figure 6.5.3 – Box plot of trade size (top) and fill rate (bottom) data for samples within each cluster.

Spread: The average spread value is defined as $\frac{1}{50}\sum_{i=1}^{50}(P_1^a(t_i-) - P_1^b(t_i-))$, where $P_1^a(t_i-)$(resp. $P_1^b(t_i-)$) is the best ask (resp. bid) price before the execution of the ith order. A low spread value indicates a more liquid market.

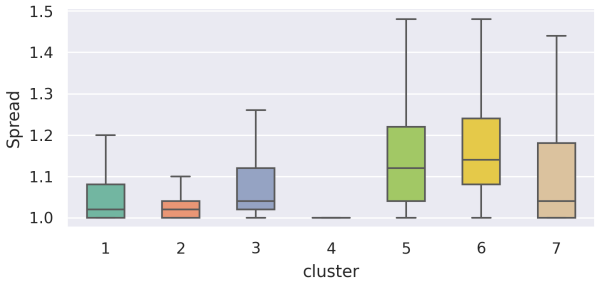


Figure 6.5.4 – Box plot of bid-ask spread immediately before trades for samples within each cluster.

Queue size (QS): The indicator **queue size** is calculated as $\frac{1}{50}\sum_{i=1}^{50} Q_1^{s_i}(t_i-)$. s_i stands for the side of the ith order (buy or sell), and $Q_1^{s_i}(t_i-)$ denotes the volume of the available orders at the best level on the same side before the ith order was executed. Some traders may choose to place a market order when the available number of limit orders is low, in order to avoid missing out on potential gains.

We also define another indicator (called **opposite queue size (opposite QS)**) as $\frac{1}{50}\sum_{i=1}^{50} Q_1^{s_i^c}(t_i-)$. s_i^c represents the opposite side of the ith order (buy or sell). $Q_1^{s_i^c}(t_i-)$ is the volume at the best level of the opposite side from where the trade occurs. For example, if order i is a buy order $s_i = a$,

$Q_1^{s_i^c}(t_i-)$ is the queue size at best bid limit. A high value of **opposite queue size** implies that if the orders were placed as limit orders, they would take a long time to be executed due to the long waiting list. We may apply this indicator to measure the level of impatience displayed by an agent, which can serve as a valuable sign of aggressive actions by a market maker.

To gain a more comprehensive understanding, we also analyzed the Related queue size (RQS) and the opposite related queue size (Opposite RQS), in addition to QS and opposite QS. RQS and opposite RQS are respectively defined by $\frac{1}{50}\sum_{i=1}^{50} Q_1^{s_i}(t_i-)/q_i$ and $\frac{1}{50}\sum_{i=1}^{50} Q_1^{s_i^c}(t_i-)/q_i$. A value of RQS close to 1 indicates that the market orders are executing almost all the best level orders. Moreover, the price is moving after these trades when $RQS = 1$.

We are inspired to include these indicators based on the observation that the arrival rate of order flows is influenced by the queue sizes. This property, named Queue Reactive, has been studied in several works Huang et al. (2015); Wu et al. (2019).

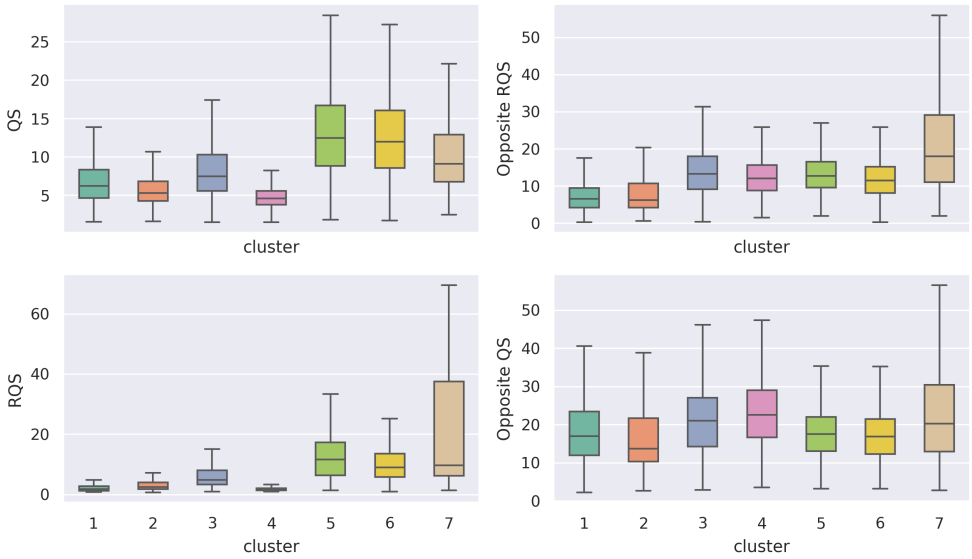


Figure 6.5.5 – Within each cluster, box plot illustrate the queue sizes of samples. On the left side, the figures represent the queue sizes at the side where trades occur, while the right side figures stand for the queue sizes on the opposite side. The top figures show the actual queue sizes, while the bottom figures display the queue sizes relative to the trade size.

Direction: We represent the direction of an input using the formula $|\sum_{i=1}^{50} q_i \cdot s_i|/(\sum_{i=1}^{50} q_i)$, where s_i is 1 if the order is a buy order, otherwise s_i is -1. We take the absolute value in the formula because the crucial information we are interested in is whether the input is directional rather than the direction itself. A direction value close to 0 indicates a balanced input, while a value close to 1 indicates a highly directional input. Specifically, a value of 0 indicates that the buy and sell orders are evenly distributed, whereas a value of 1 indicates that all orders are on the same side.

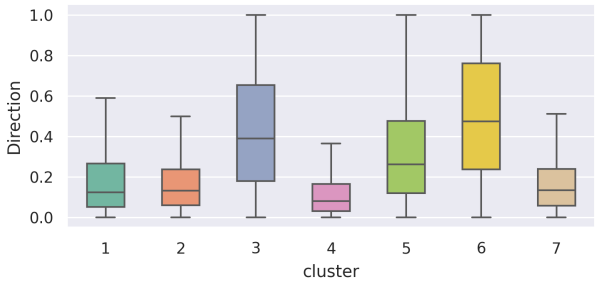


Figure 6.5.6 – Box plots of direction indicators for samples within each cluster

Limit to trade modification: The final indicator we use is the proportion of modification in all the market orders, calculated as $\frac{1}{50}\sum_{i=1}^{50} M_i$. Here M_i indicates whether the ith market order is a limit to trade modification order. Let us note that $M = 1$ means that the order was modified from an existing limit order to make it aggressive, while $M = 0$ means that the order was added aggressive.

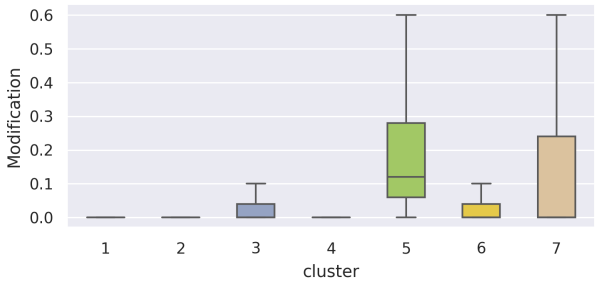


Figure 6.5.7 – Box plots of modification proportion for samples within each cluster

Based on all previous indicator analysis, we can summarize the characteristics of these clusters in the following table 6.5.1, by using a rating system ranging from (+) to (+++).

Notably, we see that

- Cluster 4 : This cluster exhibits low frequency, minimum spread, and zero modification. It is dominated by agents 10, 17, and 22. Referring to Figure 6.3.1, we observe that these three agents have almost no passive trades. Therefore in Cluster 4, agents primarily function as speculators.
- Cluster 6 : This cluster demonstrates low frequency, high bid-ask spread, and a significant directional indicator. Agents within this cluster perform directional trading.
- Cluster 7 : This cluster is characterized by a high opposite queue size, non-obvious direction and significant modification. Agent 11 and Agent 12 are the main contributors to this cluster. Examining Figure 6.3.1, we notice that these two agents exhibit a high passive-aggressive ratio, indicating that Cluster 7 represents the impatient behavior of market makers.

6.5.3 Delving into details of each agent

To further analyze the behavior of agents in different clusters, we use the indicators mentioned earlier to evaluate their samples in each cluster. We select a few agents as examples to illustrate

cluster	1	2	3	4	5	6	7
Frequency	+	+++	++	+	+	+	++
Trade size	+++	++	++	++	+	++	+
Fill rate	+	+	++	++	++	++	+++
Spread	++	+	++	+	+++	+++	++
QS	++	++	++	+	+++	+++	++
Opposite QS	+	+	++	++	++	++	+++
Direction	+	+	+++	+	++	+++	+
Modification			+		++	+	++

Table 6.5.1 – Evaluation of the clusters based on the above indicators (from none() to low (+) to high (+++)

the differences in their behavior across different clusters.

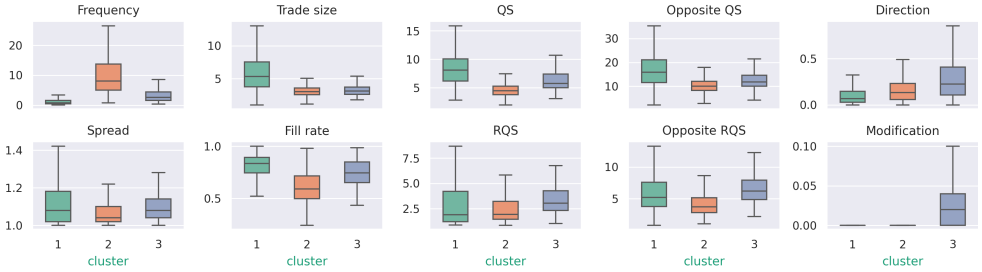
The first example is Member 9 which is, according to the results of Fig.6.5.1, assigned to clusters 1,2 and 3. Figure 6.5.8a provides insight into its behavior within these clusters. In cluster 2, Agent 9 engages in high-frequency trading, typically when the queue size is very low. When we plot the time periods of these samples throughout the trading day (as shown in Figure 6.5.8b), we observe that in the morning, Agent 9 preferentially behaves as Cluster 1, while in the afternoon, it exhibits behaviors similar to those of Cluster 2.

Another example is Agent 20, globally belongs to the clusters 3,5 and 6. We can observe that the samples of Agent 20 that belong to the cluster 3 have a higher frequency and are more likely to be active when the market is liquid as indicated by a smaller spread. In contrast, in cluster 5, with the significant modifications and lower direction index values, the agent behaves more like a market maker. (see Figure 6.5.9)

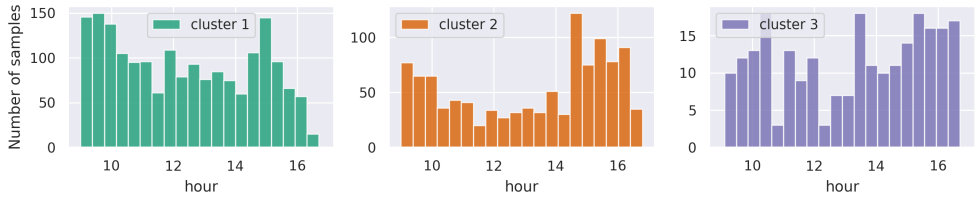
To visualize the evolution of agents' behaviors over time, we present two examples, namely agent 6 and 10, respectively in Figure 6.5.10 and 6.5.11. During the period from January 2016 to March 2017, it is observed that Agent 6 exhibits a decrease in activity while maintaining a relatively consistent behavior. On the other hand, it is observed that Agent 10 significantly changes the behavior twice during this period. The first time of change occurs around March 2016, followed by another one around December 2016.

6.5.4 Clusters visualization

A popular statistical method for visualizing high-dimensional data is the t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten and Hinton, 2008). It is a non-linear technique that maps high-dimensional data to a low-dimensional space while preserving the structure of the original data. However, in practice, t-SNE can be computationally expensive and struggle with high-dimensional data. Therefore, it is often recommended to first use another dimensionality reduction method, such as Principal component analysis (PCA), to reduce the number of dimensions to a reasonable amount before applying t-SNE. Figure 6.5.12 shows the results of applying t-SNE to 50,000 order book samples from the thirty agents.



(a)



(b)

Figure 6.5.8 – Agent 9. (a) Each figure corresponds to an indicator. Within each figure, the three vertical bars represent the performance of samples from Agent 9 within each cluster. (b) Each figure corresponds to a cluster. Within each figure, the histogram plot displays the distribution of samples selecting times.

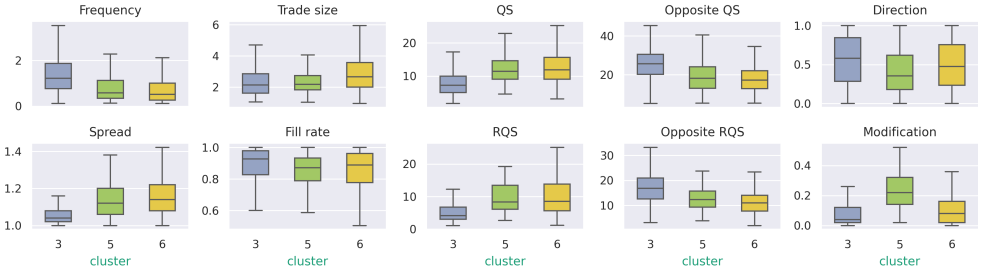
Based on the t-SNE visualization, we have observed the following noteworthy observations:

- Agent 11 and 12 are assigned to the same cluster, while their images are almost disjoint. This suggests that these two agents share similarities with respect to other agents in the dataset, but the difference between them is still distinct. Similar observations can be made for Agent 21, 24 and 26, although their dissimilarities are less clear compared to Agents 11 and 12.
- Even though Agent 1, 3, 6 and 8 all have a significant portion distributed within cluster 2, there is a higher degree of similarity among Agent 3,6 and 8 compared to the similarity between Agent 1 and the other three agents.

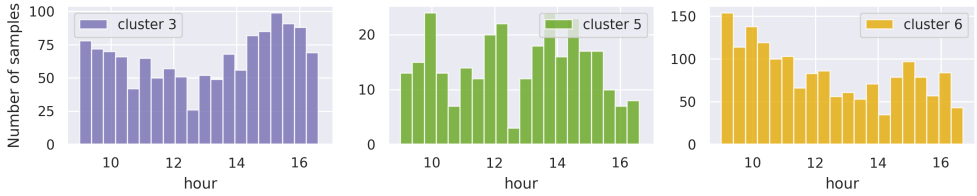
6.6 - Conclusion and Discussion

In this chapter, we present a novel approach for limit order book analysis by designing a contrastive learning method with triplet loss. Our study uses the CAC40 index future data provided by Euronext Paris, spanning from January 2016 to February 2017. We make the assumption that individual agents maintain consistent behavior over short periods, while different agents exhibit distinct behaviors. By training neural networks, we obtain vector representations of sequences of market orders from the same agent.

We employ K-means clustering on the set of obtained representation vectors, in order to group the sequences of market orders effectively. This clustering cut the set to seven clusters. Subsequently, we define various indicators such as trading frequency, spread to characterize the sequences within each cluster. This allows us to identify distinct market marker clusters as well as clusters associated with directional agents. Furthermore, we analyze the behavior of each agent across different clusters based on these indicators, offering valuable insights into their trading behavior and its evolution over time.



(a)



(b)

Figure 6.5.9 – Agent 20 : (a) Each figure corresponds to an indicator. Within each figure, the three vertical bars represent the performance of samples from Agent 20 within each cluster. (b) Each figure corresponds to a cluster. Within each figure, the histogram plot displays the distribution of samples selecting times.

In future research, we plan to expand our analysis to include both aggressive and passive trades, thus providing a more comprehensive understanding of the market. Inspired by the work Cartea et al. (2023), we also intend to extend the order features by incorporating additional factors such as deep order volume in the limit order book and agent inventories.

Furthermore, the learned representation vectors can be applied to various downstream tasks. For instance, in market forecasting, the embedding vectors from active agents can effectively represent the market context. Additionally, these vectors can be utilized for agent-based generation of synthetic market data, offering new possibilities for market simulation and analysis.

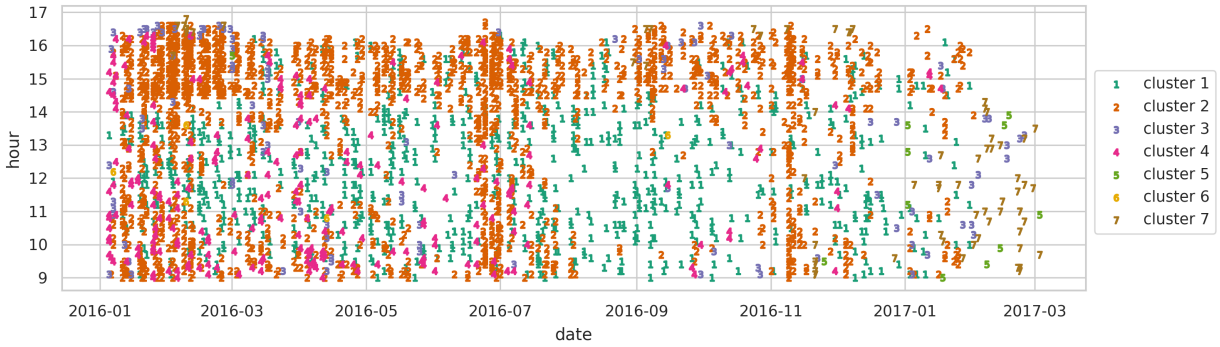


Figure 6.5.10 – 2-D scatter plot. X-axis represents the dates and the y-axis represents the hour in a day. In this plot, each point stands for the occurring time of a selected sample and its color shows the cluster that it belongs to.

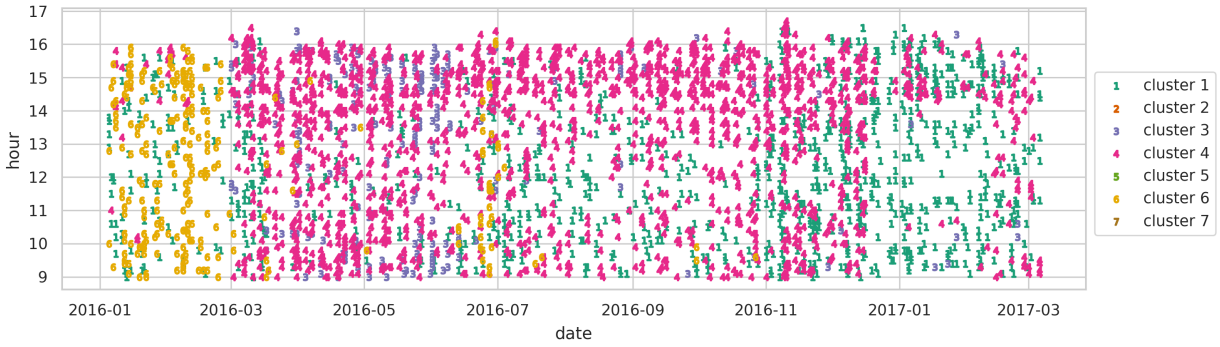


Figure 6.5.11 – 2-D scatter plot. X-axis represents the dates and the y-axis represents the hour in a day. In this plot, each point stands for the occurring time of a selected sample and its color shows the cluster that it belongs to.

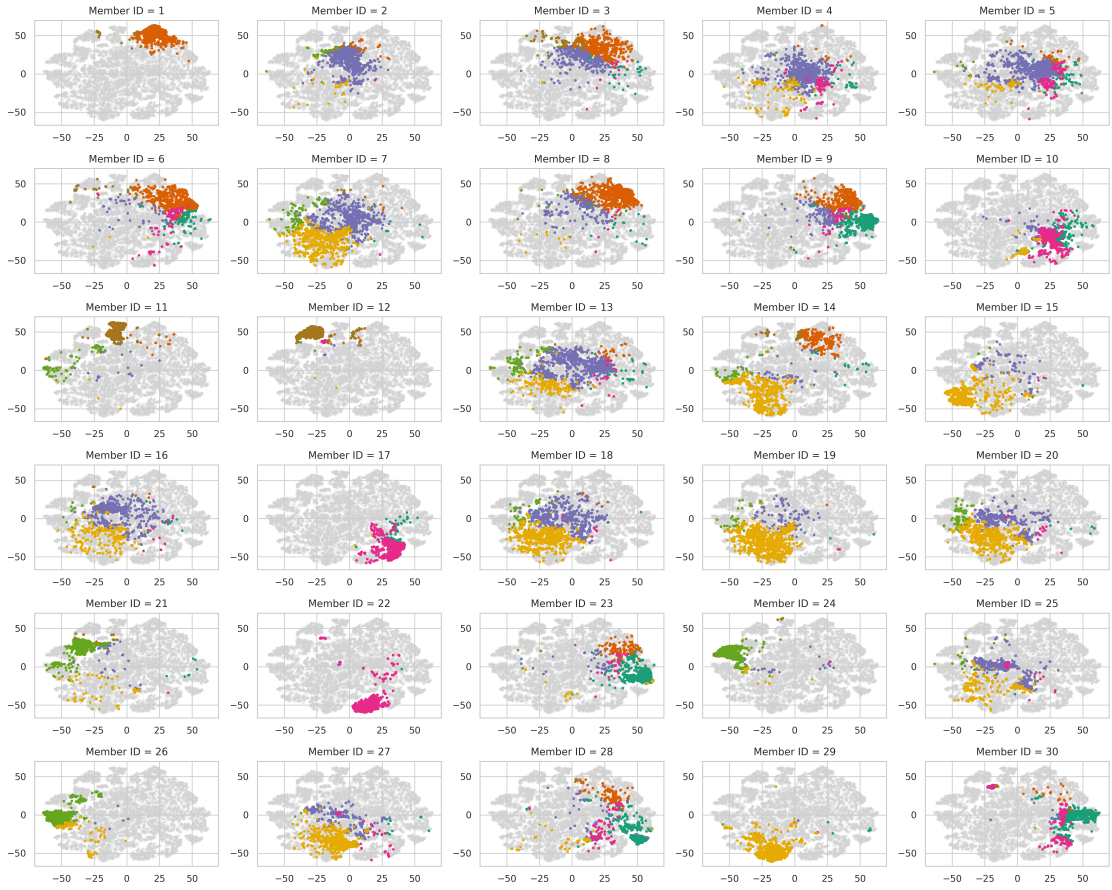


Figure 6.5.12 – t-SNE visualization of 50,000 samples : the colored parts in each subfigure represent the samples of a given agent, the different colors indicate the clusters to which the agent belongs

CONCLUSION AND PERSPECTIVES

The objective of this thesis was to analysis and model the high-frequency data in financial markets using Hawkes processes and deep learning methods. The database used in this study comprised order book data of 40 stocks as well as the futures contract of the CAC40 index, all sourced from the French Euronext market.

Before delving into the main topics, Chapter 2 was dedicated to establishing essential groundwork. This chapter provided a comprehensive introduction to topics relevant to this thesis, such as financial markets, Hawkes processes and neural networks.

The heart of this thesis was Chapters 3 to 6, where each individual chapter corresponded to a project during my Ph.D. studies. Chapter 3 was devoted to the construction of a non-linear Hawkes process model for spread dynamics, referred to as the "State-Dependent Spread Hawkes" model (SDSH). Comparing to the classical linear Hawkes model, this model integrated a "state-dependent" term into the intensity function. This term served several purposes: (i) it guaranteed the positivity of the spread value, (ii) it adjusted the intensity according to the spread state. In this model, we also allowed multiple jump sizes of the spread. With exponential kernels and some assumptions on the state-dependent term, we demonstrated the ergodicity property of this model, for a specific case. Estimations are based on the Maximum Likelihood Estimation method. We calibrated this model using the order book data of CAC40, and the results showed that the model was able to accurately reproduce diverse statistical properties of the spread. As a direction of applications, we tried to forecast the spread using this model. Comparing with other baseline models, our model showed a better performance in terms of forecasting accuracy, especially for the short-term forecast.

Chapter 4 introduced two "Hawkes process with shot noise" models, originally designed to disentangle exogenous and endogenous factors of correlation between two assets' prices. These two models are respectively named the "latent-behavior" model and the "latent-information" model. In this chapter, we concentrated mainly on the latent-behavior model. This model assumed the existence of a shot noise process, which encoded latent behaviors of agents. The latent behaviors encompassed instances where some agents are engaged in trading both assets at the same time. We proved some limit theorems for this model, by using the same techniques as in Bacry et al. (2013b). Only one section was devoted to the introduction of the latent-information model, which as well incorporated a shot noise process. However, the shot noise process in this second model represented news events. In empirical applications, we proposed to use the non parametric Hawkes

cumulant estimator. This method was proven to be effective for both the latent-behavior and latent-information models.

While the market participants analyze the market and make decisions based on price movements, regulators and exchanges are interested in the strategies and behaviors of agents. From Chapter 5, our focus shifted towards characterizing the agent behaviors by applying deep learning approaches. As each order was labeled with the agent identity who placed it, we proposed to identify an agent's behavior by sequences of consecutive orders placed by this agent. Chapter 5 employed a supervised learning method in order to categorize these sequences of orders. The numerical results on CAC40 Future data showed that the agents in the market performed differently and our model could accurately classify them. Through a comparison of different sets of features for each order, we found an optimal approach for feeding the model with the most relevant information.

Based on the work of Chapter 5, we established a more realistic and applicable task in Chapter 6. While it is not obvious to compare agent behaviors, our goal was to develop an encoder which maps agent behaviors to a Euclidean space $\mathbb{R}^n$ which is a familiar mathematical space. We applied a self-supervised learning method equipped with a triplet loss function for this purpose. During training, the encoder learned to map two similar agent behaviors to nearby points in $\mathbb{R}^n$ and two dissimilar behaviors to distant points. To clarify, in our model, a agent's behavior is represented by a sequence of market orders. Two behaviors (or two sequences of orders) were considered similar if they originated from the same agent and occurred within a close temporal proximity (less than 2 hours) while they were dissimilar if they belonged to different agents. Thanks to the acquired encoder, comparison of order sequences became possible. Subsequently, we applied the K-means clustering algorithm on the representation (in $\mathbb{R}^n$) of order sequences, which grouped order sequences (i.e., agent behaviors) into different clusters, with each cluster representing a unique trading behavior.

To summarize, our research involved the development of models based on Hawkes processes to capture Level 1 events in Limit Order Book, and the utilization of deep learning methods for the analysis and characterization of agent behaviors. We believe that our contributions have added value to both the academic and industrial communities. Nevertheless, like any scientific research, our work raises new questions and opens doors to new directions. In future studies, we can extend the previous work and explore new directions as follows.

The State-Dependent Spread Hawkes model (SDSH) can be extended in several ways. One immediate avenue for improvement lies in its application scope. In this thesis, we focused on four assets with relatively small spreads. Broadening the model's application to assets with larger mean spreads would provide a richer understanding of the "state-dependent" elements. At the same time, we aim to delve deeper in the predictive capabilities of the model, as mentioned in Chapter 3. Furthermore, there are theoretical aspects to consider, as the demonstration of stationarity for the general SDSH models is still missing. The above points are based on the existing SDSH model; of course, we can also make adjustments to the model. It is clear that the intensity of spread jumps depend on the queue size, meaning that a higher number of orders at the best levels reduces the likelihood of spread value changes. Thus, to enhance the model's capacity, we can incorporates the queue sizes into the state-dependent terms.

The development of the Hawkes process with shot noise model is an ongoing work. In Chapter 4, we focused on the the theoretical aspects of this model and its estimation using synthetic data. In future wok, we will apply this model on diverse range of assets across different markets. On the other hand, regarding estimation techniques, we can also try other methods, such as the Expectation-

Maximization (EM) algorithm. In Section 4.C, we provided an overview of the potential application of the Sequential Monte Carlo EM algorithm. However, it's important to note that we have not yet implemented this algorithm nor established its convergence properties. This remains a subject of our ongoing research efforts.

As to the deep learning methods for analyzing agent behaviors, our approach will continue to build upon the techniques outlined in Chapter 6. In our future research, we have plans to expand our input data by incorporating additional features, including: (i) providing both aggressive and passive trades, (ii) integrating deeper order volume and agent inventories, (iii) expanding our scope to include a broader range of agents, rather than just the market takers. Furthermore, the learned representation vectors of agent behaviors can be applied to a wide range of downstream tasks. Beyond these applications, we can also address more challenging problems related to agent behaviors in the market. At the macroscopic level, we want forecast the trading volume of agents, or predict whether they are net buyers or net sellers. At the microscopic level, we can try to predict the next order of an agent, including its price level, direction (buy or sell), and timing, based on their previous order history.

BIBLIOGRAPHY

Abergel, F., Anane, M., Chakraborti, A., Jedidi, A., and Toke, I. M. (2016). *Limit order books*. Cambridge University Press.

Abergel, F. and Jedidi, A. (2013). A mathematical approach to order book modeling. *International Journal of Theoretical and Applied Finance*, 16(05):1350025.

Abergel, F. and Jedidi, A. (2015). Long-time behavior of a hawkes process–based limit order book. *SIAM Journal on Financial Mathematics*, 6(1):1026–1043.

Achab, M., Bacry, E., Gaıffas, S., Mastromatteo, I., and Muzy, J.-F. (2017). Uncovering causality from multivariate hawkes integrated cumulants. In *International Conference on Machine Learning*, pages 1–10. PMLR.

Achab, M., Bacry, E., Muzy, J.-F., and Rambaldi, M. (2018). Analysis of order book flows using a non-parametric estimation of the branching ratio matrix. *Quantitative Finance*, 18(2):199–212.

Axtell, R. L. and Farmer, J. D. (2022). Agent-based modeling in economics and finance: Past, present, and future. *Journal of Economic Literature*.

Bacry, E., Bompaire, M., Gaïffas, S., and Poulsen, S. (2017). Tick: a python library for statistical learning, with a particular emphasis on time-dependent modelling. *arXiv preprint arXiv:1707.03003*.

Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J.-F. (2013a). Modelling microstructure noise with mutually exciting point processes. *Quantitative finance*, 13(1):65–77.

Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J.-F. (2013b). Some limit theorems for hawkes processes and application to financial statistics. *Stochastic Processes and their Applications*, 123(7):2475–2499.

Bacry, E., Jaisson, T., and Muzy, J.-F. (2016). Estimation of slowly decreasing hawkes kernels: application to high-frequency order book dynamics. *Quantitative Finance*, 16(8):1179–1201.

Bacry, E., Mastromatteo, I., and Muzy, J.-F. (2015). Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005.

Bacry, E. and Muzy, J.-F. (2014). Hawkes model for price and trades high-frequency dynamics. *Quantitative Finance*, 14(7):1147–1166.

Bacry, E. and Muzy, J.-F. (2016). First-and second-order statistics characterization of hawkes processes and non-parametric estimation. *IEEE Transactions on Information Theory*, 62(4):2184–2202.

Bessembinder, H. (1994). Bid-ask spreads in the interbank foreign exchange markets. *Journal of Financial economics*, 35(3):317–348.

Biais, B., Foucault, T., et al. (2014). Hft and market quality. *Bankers, Markets & Investors*, 128(1):5–19.

Bochud, T. and Challet, D. (2007). Optimal approximations of power laws with exponentials: application to volatility models with long memory. *Quantitative Finance*, 7(6):585–589.

Bouchaud, J.-P. (2011). The endogenous dynamics of markets: price impact, feedback loops and instabilities. *Lessons from the 2008 Crisis*, pages 57–156.

Bouchaud, J.-P., Bonart, J., Donier, J., and Gould, M. (2018). *Trades, quotes and prices: financial markets under the microscope*. Cambridge University Press.

Bouchaud, J.-P., Farmer, J. D., and Lillo, F. (2009). How markets slowly digest changes in supply and demand. In *Handbook of financial markets: dynamics and evolution*, pages 57–160. Elsevier.

Bouchaud, J.-P., Mézard, M., and Potters, M. (2002). Statistical properties of stock order books: empirical results and models. *Quantitative finance*, 2(4):251.

Bowser, C. (2007). Modelling security market events in continuous time: Intensity based point process models. *Journal of Econometrics*, 141(2):876–912.

Bowsher, C. G. (2007). Modelling security market events in continuous time: Intensity based, multivariate point process models. *Journal of Econometrics*, 141(2):876–912.

Brémaud, P. and Massoulié, L. (1996). Stability of nonlinear hawkes processes. *The Annals of Probability*, pages 1563–1588.

Briola, A., Turiel, J., and Aste, T. (2020). Deep learning modeling of the limit order book: A comparative perspective. *Available at SSRN 3714230*.

Brogaard, J. et al. (2010). High frequency trading and its impact on market quality. *Northwestern University Kellogg School of Management Working Paper*, 66.

Brogaard, J., Hendershott, T., and Riordan, R. (2014). High-frequency trading and price discovery. *The Review of Financial Studies*, 27(8):2267–2306.

Cappé, O. (2009). Online sequential monte carlo em algorithm. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 37–40. IEEE.

Cartea, Á., Cohen, S. N., Graumans, R., Labyad, S., Sánchez-Betancourt, L., and van Veldhuijzen, L. (2023). Statistical predictions of trading strategies in electronic markets. *Available at SSRN 4442770*.

Cartea, Á., Cohen, S. N., and Labyad, S. (2021). Gradient-based estimation of linear hawkes processes with general kernels. *arXiv preprint arXiv:2111.10637*.

Cattivelli, L. and Pirino, D. (2019). A sharp model of bid–ask spread forecasts. *International Journal of Forecasting*, 35(4):1211–1225.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chiang, W.-H., Liu, X., and Mohler, G. (2022). Hawkes process modeling of covid-19 with mobility leading indicators and spatial covariates. *International journal of forecasting*, 38(2):505–520.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F. et al. (2018). Keras: The python deep learning library. *Astrophysics source code library*, pages ascl–1806.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Cont, R., Cucuringu, M., Glukhov, V., and Prenzel, F. (2023). Analysis and modeling of client order flow in limit order markets. *Quantitative Finance*, pages 1–19.

Cont, R., Stoikov, S., and Talreja, R. (2010). A stochastic model for order book dynamics. *Operations research*, 58(3):549–563.

Da Fonseca, J. and Zaatour, R. (2015). Clustering and mean reversion in a hawkes microstructure model. *Journal of Futures Markets*, 35(9):813–838.

Da Fonseca, J. and Zaatour, R. (2017). Correlation and lead–lag relationships in a hawkes microstructure model. *Journal of Futures Markets*, 37(3):260–285.

Daley, D. J., Vere-Jones, D., et al. (2003). *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer.

Daniels, M. G., Farmer, J. D., Gillemot, L., Iori, G., and Smith, E. (2003). Quantitative model of price diffusion and market friction based on trading as a mechanistic random process. *Physical review letters*, 90(10):108102.

Dassios, A. and Zhao, H. (2013). Exact simulation of hawkes process with exponentially decaying intensity. *Electronic Communications in Probability*, 18:1–13.

Dayri, K. and Rosenbaum, M. (2015). Large tick assets: implicit spread and optimal tick size. *Market Microstructure and Liquidity*, 1(01):1550003.

de Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22.

Díaz, A. and Escribano, A. (2020). Measuring the multi-faceted dimension of liquidity in financial markets: A literature review. *Research in International Business and Finance*, 51:101079.

Dixon, M. (2018). Sequence classification of the limit order book using recurrent neural networks. *Journal of computational science*, 24:277–286.

Doering, J., Fairbank, M., and Markose, S. (2017). Convolutional neural networks applied to high-frequency market microstructure forecasting. In *2017 9th computer science and electronic engineering (ceec)*, pages 31–36. IEEE.

Down, D., Meyn, S. P., and Tweedie, R. L. (1995). Exponential and uniform ergodicity of markov processes. *The Annals of Probability*, 23(4):1671–1691.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. (2016). Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1555–1564.

Fall, M., Louhichi, W., and Viviani, J. L. (2021). Forecasting the intra-day effective bid ask spread by combining density forecasts. *Applied Economics*, 53(50):5772–5792.

Farmer, J. D., Patelli, P., and Zovko, I. I. (2005). The predictive power of zero intelligence in financial markets. *Proceedings of the National Academy of Sciences*, 102(6):2254–2259.

Fong, K. Y., Holden, C. W., and Trzcinka, C. A. (2017). What are the best liquidity proxies for global research? *Review of Finance*, 21(4):1355–1401.

Fosset, A., Bouchaud, J.-P., and Benzaquen, M. (2020a). Endogenous liquidity crises. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(6):063401.

Fosset, A., Bouchaud, J.-P., and Benzaquen, M. (2020b). Non-parametric estimation of quadratic hawkes processes for order book events. *Available at SSRN 3599027*.

Foucault, T., Kadan, O., and Kandel, E. (2005). Limit order book as a market for liquidity. *The review of financial studies*, 18(4):1171–1217.

Garetto, M., Leonardi, E., and Torrisi, G. L. (2021). A time-modulated hawkes process to model the spread of covid-19 and the impact of countermeasures. *Annual reviews in control*, 51:551–563.

Glosten, L. R. and Harris, L. E. (1988). Estimating the components of the bid/ask spread. *Journal of financial Economics*, 21(1):123–142.

Glosten, L. R. and Milgrom, P. R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of financial economics*, 14(1):71–100.

Gould, M. D., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J., and Howison, S. D. (2013). Limit order books. *Quantitative Finance*, 13(11):1709–1742.

Goyenko, R. Y., Holden, C. W., and Trzcinka, C. A. (2009). Do liquidity measures measure liquidity? *Journal of financial Economics*, 92(2):153–181.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Groß-KlußMann, A. and Hautsch, N. (2013). Predicting bid–ask spreads using long-memory autoregressive conditional poisson models. *Journal of Forecasting*, 32(8):724–742.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Hagströmer, B. and Nordén, L. (2013). The diversity of high-frequency traders. *Journal of Financial Markets*, 16(4):741–770.

Hardiman, S. J., Bercot, N., and Bouchaud, J.-P. (2013). Critical reflexivity in financial markets: a hawkes process analysis. *Eur. Phys. J. B*, 86:442.

Harris, L. (2003). *Trading and exchanges: Market microstructure for practitioners*. OUP USA.

Hasbrouck, J. (2007). *Empirical market microstructure: The institutions, economics, and econometrics of securities trading*. Oxford University Press.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hawkes, A. G. (1971a). Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443.

Hawkes, A. G. (1971b). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90.

Hawkes, A. G. (1973). Cluster models for earthquakes-regional comparisons. *Bull. Int. Stat. Inst.*, 45(3):454–461.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hou, M., Xu, C., Liu, Y., Liu, W., Bian, J., Wu, L., Li, Z., Chen, E., and Liu, T.-Y. (2021). Stock trend prediction with multi-granularity data: A contrastive learning approach with adaptive fusion. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 700–709.

Huang, R. D. and Stoll, H. R. (1997). The components of the bid-ask spread: A general approach. *The Review of Financial Studies*, 10(4):995–1034.

Huang, W., Lehalle, C.-A., and Rosenbaum, M. (2015). Simulating and analyzing order book data: The queue-reactive model. *Journal of the American Statistical Association*, 110(509):107–122.

Iori, G. and Porter, J. (2018). Agent-based modeling for financial markets.

Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, 184:115537.

Jovanović, S., Hertz, J., and Rotter, S. (2015). Cumulants of hawkes point processes. *Physical Review E*, 91(4):042802.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirchner, M. (2017). An estimation procedure for the hawkes process. *Quantitative Finance*, 17(4):571–595.

Kirilenko, A., Kyle, A. S., Samadi, M., and Tuzun, T. (2017). The flash crash: High-frequency trading in an electronic market. *The Journal of Finance*, 72(3):967–998.

Kobayashi, R. and Lambiotte, R. (2016). Tideh: Time-dependent hawkes process for predicting retweet dynamics. In *Tenth International AAAI Conference on Web and Social Media*.

Kumbure, M. M., Lohrmann, C., Luukka, P., and Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, page 116659.

Lambert, R. C., Tuleau-Malot, C., Bessaih, T., Rivoirard, V., Bouret, Y., Leresche, N., and Reynaud-Bouret, P. (2018). Reconstructing the functional connectivity of multiple spike trains using hawkes models. *Journal of neuroscience methods*, 297:9–21.

Large, J. (2007). Measuring the resiliency of an electronic limit order book. *Journal of Financial Markets*, 10(1):1–25.

Laub, P. J., Lee, Y., and Taimre, T. (2021). *The elements of Hawkes processes*. Springer.

Lawrance, A. and Lewis, P. A. (1975). Properties of the bivariate delayed poisson process. *Journal of Applied Probability*, 12(2):257–268.

Lee, K. and Seo, B. K. (2022). Modeling Bid and Ask Price Dynamics with an Extended Hawkes Process and Its Empirical Applications for High-Frequency Stock Market Data. *Journal of Financial Econometrics*. nbab029.

Lehalle, C.-A. and Laruelle, S. (2018). *Market microstructure in practice*. World Scientific.

Lewis, E. and Mohler, G. (2011). A nonparametric em algorithm for multiscale hawkes processes. *Journal of Nonparametric Statistics*, 1(1):1–20.

Lewis, P. (1972). Multivariate point processes. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, page 401. University of California Press.

Lewis, P. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Linderman, S. W., Wang, Y., and Blei, D. M. (2017). Bayesian inference for latent hawkes processes. *Advances in Neural Information Processing Systems*.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297.

Maglaras, C., Moallemi, C. C., and Wang, M. (2022). A deep learning approach to estimating fill probabilities in a limit order book. *Quantitative Finance*, 22(11):1989–2003.

Mäkinen, Y., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2019). Forecasting jump arrivals in stock prices: new attention-based network architecture using limit order book data. *Quantitative Finance*, 19(12):2033–2050.

Marcaccioli, R., Bouchaud, J.-P., and Benzaquen, M. (2022). Exogenous and endogenous price jumps belong to different dynamical classes. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(2):023403.

Mehari, T. and Strodthoff, N. (2022). Self-supervised representation learning from 12-lead ecg data. *Computers in biology and medicine*, 141:105114.

Mei, H. and Eisner, J. M. (2017). The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30.

Mei, H., Qin, G., and Eisner, J. (2019). Imputing missing events in continuous-time event streams. In *International Conference on Machine Learning*, pages 4475–4485. PMLR.

Mike, S. and Farmer, J. D. (2008). An empirical behavioral model of liquidity and volatility. *Journal of Economic Dynamics and Control*, 32(1):200–234.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mohsenvand, M. N., Izadi, M. R., and Maes, P. (2020). Contrastive representation learning for electroencephalogram classification. In *Machine Learning for Health*, pages 238–253. PMLR.

Morariu-Patrichi, M. and Pakkanen, M. S. (2022). State-dependent hawkes processes and their application to limit order book modelling. *Quantitative Finance*, 22(3):563–583.

Muni Toke, I. and Pomponio, F. (2012). Modelling trades-through in a limit order book using hawkes processes. *Economics - The Open-Access, Open-Assessment E-Journal*, 6:1–23.

Muni Toke, I. and Yoshida, N. (2017). Modelling intensities of order flows in a limit order book. *Quantitative Finance*, 17(5):683–701.

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., and Salwana, E. (2020). Deep learning for stock market prediction. *Entropy*, 22(8):840.

Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2018). Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37(8):852–866.

Ogata, Y. (1981). On lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31.

Ogata, Y. and Akaike, H. (1982). On linear intensity models for mixed doubly stochastic poisson and self-exciting point processes. In *Selected Papers of Hirotugu Akaike*, pages 269–274. Springer.

O'hara, M. (1998). *Market microstructure theory*. John Wiley & Sons.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Plerou, V., Gopikrishnan, P., and Stanley, H. E. (2005). Quantifying fluctuations in market liquidity: Analysis of the bid-ask spread. *Physical Review E*, 71(4):046131.

Ponzi, A., Lillo, F., and Mantegna, R. N. (2006). Market reaction to temporary liquidity crises and the permanent market impact. *arXiv preprint physics/0608032*.

Rambaldi, M., Bacry, E., and Lillo, F. (2017). The role of volume in order book dynamics: a multivariate hawkes process analysis. *Quantitative Finance*, 17(7):999–1020.

Rambaldi, M., Bacry, E., and Muzy, J.-F. (2019). Disentangling and quantifying market participant volatility contributions. *Quantitative Finance*, 19(10):1613–1625.

Rasmussen, J. G. (2013). Bayesian inference for hawkes processes. *Methodology and Computing in Applied Probability*, 15:623–642.

Reynaud-Bouret, P., Rivoirard, V., and Tuleau-Malot, C. (2013). Inference of functional connectivity in neurosciences via hawkes processes. In *2013 IEEE global conference on signal and information processing*, pages 317–320. IEEE.

Reynaud-Bouret, P. and Schbath, S. (2010). Adaptive estimation for hawkes processes; application to genome analysis.

Rizoiu, M.-A., Lee, Y., Mishra, S., and Xie, L. (2017). A tutorial on hawkes processes for events in social media. *arXiv preprint arXiv:1708.06401*.

Rizoiu, M.-A., Mishra, S., Kong, Q., Carman, M., and Xie, L. (2018). Sir-hawkes: Linking epidemic models and hawkes processes to model diffusions in finite populations. In *Proceedings of the 2018 world wide web conference*, pages 419–428.

Roşu, I. (2009). A dynamic model of the limit order book. *The Review of Financial Studies*, 22(11):4601–4641.

Ruan, R., Bacry, E., and Muzy, J.-F. (2023a). Liquidity takers behavior representation through a contrastive learning approach. In *ICAIF-23*.

Ruan, R., Bacry, E., and Muzy, J.-F. (2023b). The self-exciting nature of the bid-ask spread dynamics. *arXiv preprint arXiv:2303.02038*.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Sfendourakis, E. and Toke, I. M. (2021). Lob modeling using hawkes processes with a state-dependent factor. *arXiv preprint arXiv:2107.12872*.

Shelton, C., Qin, Z., and Shetty, C. (2018). Hawkes process inference with missing data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Sirignano, J. and Cont, R. (2019). Universal features of price formation in financial markets: perspectives from deep learning. *Quantitative Finance*, 19(9):1449–1459.

Sirignano, J. A. (2019). Deep learning for limit order books. *Quantitative Finance*, 19(4):549–570.

Smith, E., Farmer, J. D., Gillemot, L., and Krishnamurthy, S. (2003). Statistical theory of the continuous double auction. *Quantitative finance*, 3(6):481.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Sornette, D. (2006). Endogenous versus exogenous origins of crises. In *Extreme events in nature and society*, pages 95–119. Springer.

Sornette, D. and Helmstetter, A. (2003). Endogenous versus exogenous shocks in systems with memory. *Physica A: Statistical Mechanics and its Applications*, 318(3-4):577–591.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Stoll, H. R. (1989). Inferring the components of the bid-ask spread: Theory and empirical tests. *the Journal of Finance*, 44(1):115–134.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Tashiro, D., Matsushima, H., Izumi, K., and Sakaji, H. (2019). Encoding of high-frequency order information and prediction of short-term stock price by deep learning. *Quantitative Finance*, 19(9):1499–1506.

Toke, I. M. (2011). "market making" in an order book model and its impact on the spread. In *Econophysics of Order-driven Markets: Proceedings of Econophys-Kolkata V*, pages 49–64. Springer.

Toke, I. M. and Yoshida, N. (2016). Modelling intensities of order flows in a limit order book, quantitative finance. *To appear,(2017)(online Version 2016)*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Veen, A. and Schoenberg, F. P. (2008). Estimation of space–time branching process models in seismology using an em–type algorithm. *Journal of the American Statistical Association*, 103(482):614–624.

Wallbridge, J. (2020). Transformers for limit order books. *arXiv preprint arXiv:2003.00130*.

Wang, J., Sun, T., Liu, B., Cao, Y., and Wang, D. (2018). Financial markets prediction with deep learning. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 97–104. IEEE.

Wu, H., Gattami, A., and Flierl, M. (2020). Conditional mutual information-based contrastive loss for financial time series forecasting. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–7.

Wu, P., Rambaldi, M., Muzy, J.-F., and Bacry, E. (2019). Queue-reactive hawkes models for the order flow. *arXiv e-prints*, pages arXiv–1901.

Wyart, M., Bouchaud, J.-P., Kockelkoren, J., Potters, M., and Vettorazzo, M. (2008). Relation between bid–ask spread, impact and volatility in order-driven markets. *Quantitative finance*, 8(1):41–57.

Yan, B., Aasma, M., et al. (2020). A novel deep learning framework: Prediction and analysis of financial time series using ceemd and lstm. *Expert systems with applications*, 159:113609.

Zawadowski, Á. G., Andor, G., and Kertész, J. (2006). Short-term market reaction after extreme price changes of liquid stocks. *Quantitative Finance*, 6(4):283–295.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR.

Zhang, Q., Lipani, A., Kirnap, O., and Yilmaz, E. (2020). Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR.

Zhang, Z., Lim, B., and Zohren, S. (2021). Deep learning for market by order data. *Applied Mathematical Finance*, 28(1):79–95.

Zhang, Z., Zohren, S., and Roberts, S. (2018). Bdlob: Bayesian deep convolutional neural networks for limit order books. *arXiv preprint arXiv:1811.10041*.

Zhang, Z., Zohren, S., and Roberts, S. (2019). Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012.

Zheng, B., Roueff, F., and Abergel, F. (2014). Modelling bid and ask prices using constrained hawkes processes: Ergodicity and scaling limit. *SIAM Journal on Financial Mathematics*, 5(1):99–136.

Zhu, L. (2013). *Nonlinear Hawkes processes*. PhD thesis, New York University.

Zumbach, G. (2004). How trading activity scales with company size in the ftse 100. *Quantitative finance*, 4(4):441–456.

Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. (2020). Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR.

RÉSUMÉ

Cette thèse est consacrée à l'étude de la microstructure du marché dans les marchés électroniques, en mettant l'accent sur deux sujets clés. Le premier sujet concerne la construction de deux modèles pour les événements de Niveau 1 dans le carnet d'ordres, en utilisant des approches basées sur des modèles statistiques. Le premier modèle consiste en un processus de Hawkes non-linéaire pour modéliser la dynamique du *bid-ask spread*, appelé le modèle "*State-Dependent Spread Hawkes*". En intégrant les tailles des sauts du *spread* et sa valeur dans la fonction d'intensité, ce modèle est capable de capturer diverses propriétés statistiques du *spread*. Le second modèle, appelé "*Hawkes process with shot noise*", est utilisé pour séparer les sources de corrélation endogènes et exogènes entre deux prix d'actifs. Pour ce faire, ce modèle suppose l'existence d'un processus latent (*shot noise*), représentant des comportements d'agents spécifiques non directement observables sur le marché. Théoriquement, des théorèmes de limite sont démontrés et dans la pratique, l'estimation est facilitée par une technique d'estimation non paramétrique.
Le second sujet concerne l'analyse et la caractérisation des comportements des agents sur le marché financier, en utilisant des approches basées sur des réseaux neuronaux profonds. Ce sujet comprend deux tâches. La première tâche consiste à classifier les agents en fonction de leurs ordres passés, grâce à une approche d'apprentissage supervisé. La deuxième tâche vise à apprendre la représentation des comportements des agents, en utilisant un modèle d'apprentissage auto-supervisé fondé sur la *triplet loss*. Ces représentations apprises nous permettent d'appliquer l'algorithme de clustering K-means pour identifier des types de comportements distincts au sein de chaque groupe et ainsi analyser les comportements des agents.

MOTS CLÉS

Microstructure du marché, Carnet d'ordres, Processus de Hawkes, Réseaux neuronaux

ABSTRACT

This thesis is devoted to the study of market microstructure in electronic markets, focusing on two key topics. The first topic concerns the construction of two models for Level 1 events in Limit Order Book, using model-driven approches. The first model is a non-linear Hawkes process for modeling spread dynamics, referred to as the "State-Dependent Spread Hawkes" model. This model, integrating spread jump sizes and spread state into intensity, can capture a range of statistical properties of the spread. The second model, called the "Hawkes process with shot noise" model, is used to disentangle the endogenous and exogenous sources of correlation between two asset prices. To do so, this model assumes the existence of a latent shot noise process, representing specific agent behaviors not directly observable in the market. Theoretically, limit theorems are demonstrated and in practice, the estimation is facilitated through a non-parametric technique.
The second topic involves analysis and characterization of agent behaviors in the financial market, by employing data-driven approaches that relies on deep neural networks. This topic includes two tasks. The first task is to classify agents, based on their placed orders, through a supervised learning approach. The second task is to learn the representation of agents' behaviors, using a self-supervised learning model based on Triplet loss. These learning representations allow us to apply the K-means clustering algorithm to identify distinct behavior types within each cluster and therefore analyze the behaviors of agents.

KEYWORDS

Market microstructure, Limit Order Book, Hawkes processes, Neural networks