PhD-FSTM-2022-051
The Faculty of Sciences, Technology and Medicine

Université de Lorraine
Faculté de sciences, technologies et sciences de l'ingénieur

DISSERTATION

Defence held on 26/04/2022 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

AND

DOCTEUR DE L'UNIVERSITÉ DE LORRAINE

EN MATHÉMATIQUES

by

Laurent LESAGE
Born on 20th January 1991 in Mont-Saint-Martin (France)

DATA ANALYSIS FOR INSURANCE: RECOMMENDATION SYSTEM BASED ON A MULTIVARIATE HAWKES PROCESS

Dissertation defence committee

Dr Ulrich SORGER, Chair
Professor, Université du Luxembourg

Dr Caroline HILLAIRET, Vice-chair
Professor, ENSAE-Paris, CREST

Dr Radu STATE, supervisor
Professor, Université du Luxembourg

Dr Madalina DEACONU, co-supervisor
Professor, Université de Lorraine

Dr Anthony REVEILLAC
Professor, INSA Toulouse

Dr Antoine Lejay
Professor, Université de Lorraine

Remerciements

Dans un premier temps, merci à l'ensemble des trois acteurs principaux impliqués dans cette thèse, dont la multidisciplinarité a permis à nos travaux de proposer des contributions tant sur des aspects théoriques que plus appliqués. Tout d'abord, je souhaiterais remercier l'ensemble des membres de l'Université du Luxembourg que j'ai eu le privilège de côtoyer. Merci à mon co-directeur de thèse Radu State pour son encadrement tout au long de la thèse et pour m'avoir transmis ses connaissances, son expérience et son expertise en termes de Machine Learning appliqué à des problématiques industrielles. Merci à Jorge Augusto Meira pour les travaux menés ensemble lors de la rédaction des différentes publications constituant le support principal de cette thèse. Un grand merci à tout le personnel administratif que j'ai eu l'occasion d'embêter au fil de l'eau : Catherine pour son aide concernant les formations, Valérie pour l'organisation de la soutenance et le reste, Jessica pour toutes les démarches administratives.
Merci ensuite à ma co-directrice de thèse Madalina Deaconu et à Antoine Lejay pour leur supervision, pour leur apport sur les aspects théoriques de la thèse, pour leur expertise quant à la mise en valeur scientifique de nos travaux et pour leur suivi très rigoureux de l'avancée de la thèse. Merci à l'ensemble de l'équipe PASTA de l'IECL pour les échanges autour des processus ponctuels qui ont enrichi nos travaux, et pour les séminaires scientifiques de qualité.
Enfin, un grand merci à Foyer Assurances de m'avoir permis de vivre une vraie expérience professionnelle, en conciliant la progression de la thèse avec les besoins industriels et en me permettant de participer aux nombreux événements liés à l'innovation proposés par l'entreprise. Merci à mon superviseur Geoffrey Nichil pour sa coordination globale et mon implication au sein du Data Studio, ainsi qu'aux autres membres de l'équipe pour cette expérience très enrichissante (François, Alexandre, Loic, tout le monde !) et à l'équipe innovation (Marc et Margaux).

Dans un second temps, je tiens à remercier l'ensemble des personnes m'ayant fait l'honneur de participer à mon jury : Ulrich Sorger en tant que président, ainsi que les deux rapporteurs Anthony Reveillac et Caroline Hillairet, que je remercie sincèrement pour la qualité de leur rapport qui a permis d'apporter des améliorations significatives au manuscrit.

Dans un troisième temps, je remercie le Fonds National de la Recherche luxembourgeois pour son soutien financier qui a permis d'enrichir la qualité de nos travaux à travers la participation à des séminaires, conférences et formations.

Dans un quatrième temps, je remercie tous ceux qui se sont intéressés à nos travaux (ou ont essayé de le faire) et m'ont soutenu dans ce projet : ma famille au complet, en particulier mes parents qui m'ont toujours soutenu humainement et matériellement dans tous mes choix d'études, ma seconde famille constituée par mes amis proches, ma femme et mes chiens.

Abstract

The objective of the thesis is to build a recommendation system for insurance. By observing the behaviour and the evolution of a customer in the insurance context, customers seem to modify their insurance cover when a significant event happens in their life. In order to take into account the influence of life events (e.g. marriage, birth, change of job) on the insurance covering selection from customers, we model the recommendation system with a Multivariate Hawkes Process (MHP), which includes several specific features aiming to compute relevant recommendations to customers from a Luxembourgish insurance company.

Several of these features are intent to propose a personalized background intensity for each customer thanks to a Machine Learning model, to use triggering functions suited for insurance data or to overcome flaws in real-world data by adding a specific penalization term in the objective function. We define a complete framework of Multivariate Hawkes Processes with a Gamma density excitation function (i.e. estimation, simulation, goodness-of-fit) and we demonstrate some mathematical properties (i.e. expectation, variance) about the transient regime of the process. Our recommendation system has been back-tested over a full year. Observations from model parameters and results from this back-test show that taking into account life events by a Multivariate Hawkes Process allows us to improve significantly the accuracy of recommendations.

The thesis is presented in five chapters. Chapter 1 explains how the background intensity of the Multivariate Hawkes Process is computed thanks to a Machine Learning algorithm, so that each customer has a personalized recommendation. Chapter 1 is shown an extended version of the method presented in [1], in which the method is used to make the algorithm explainable. Chapter 2 presents a Multivariate Hawkes Processes framework in order to compute the dependency between the propensity to accept a recommendation and the occurrence of life events: definitions, notations, simulation, estimation, properties, etc. Chapter 3 presents several results of the recommendation system: estimated parameters of the model, effects of contributions, back-testing of the model's accuracy, etc. Chapter 4 presents the implementation of our work into a R package. Chapter 5 concludes on the contributions and perspectives opened by the thesis.

Keywords: Recommendation System, Multivariate Hawkes Processes, Insurance, Life Events, Point Processes, Machine Learning, Up-Selling, XGBoost Algorithm.

Résumé

L'objectif de la thèse est de construire un moteur de recommandation pour l'assurance. En observant leurs comportements et leur parcours dans le contexte assurantiel, les clients semblent faire évoluer leur couverture d'assurance lorsqu'un événement significatif survient dans leur vie. Afin de prendre en compte l'influence des événements de vie (mariage, naissance, nouvel emploi, etc.) sur la sélection des garanties d'assurance pour des clients, nous modélisons un moteur de recommandation avec un processus de Hawkes multivarié, qui inclut plusieurs spécificités visant à calculer des recommandations pertinentes aux clients d'une compagnie d'assurance luxembourgeoise.

Plusieurs de ces spécificités visent à proposer une « background intensity » personnalisée pour chaque client grâce à un modèle de Machine Learning, à utiliser des fonctions de déclenchement adaptées aux données assurantielles ou à pallier les failles des données réelles des clients en ajoutant un terme de pénalisation spécifique dans la fonction objectif servant à l'apprentissage des paramètres. Nous définissons un cadre complet de processus de Hawkes multivariés avec une fonction de déclenchement de densité gamma (c'est-à-dire estimation, simulation, qualité de l'ajustement) et nous démontrons certaines propriétés mathématiques (espérance, variance) sur le régime transitoire du processus. Notre moteur de recommandation a été back-testé sur une année entière. Les observations des paramètres du modèle et les résultats de ce back-test montrent que la prise en compte des événements de la vie par un processus de Hawkes multivarié permet d'améliorer significativement la précision des recommandations.

La thèse est présentée en cinq chapitres. Le chapitre 1 explique comment la « background intensity » du processus de Hawkes multivarié est calculée grâce à un algorithme de Machine Learning, afin que chaque client soit le sujet d'une recommandation personnalisée. Le chapitre 1 présente une version étendue de la méthode présentée dans [1], rendant l'algorithme interprétable. Le chapitre 2 présente un cadre de processus de Hawkes multivariés afin de calculer la dépendance entre la propension à accepter une recommandation et l'occurrence d'événements de la vie : définitions, notations, simulation, estimation, propriétés, etc. Le chapitre 3 présente plusieurs résultats du système de recommandation : paramètres estimés du modèle, effets des contributions, back-testing de la précision du modèle, etc. Le chapitre 4 présente l'implémentation de notre travail dans un package R. Le chapitre 5 conclut sur les contributions et les perspectives ouvertes par la thèse.

Mots-clés: Moteur de recommandation, processus de Hawkes multivarié, assurance, événements de vie, processus ponctuels, Machine Learning, up-selling, algorithme XGBoost.

List of papers and preprints being part of this thesis

- L. Lesage, M. Deaconu, A. Lejay, J. Meira, G. Nichil, and R. State, "A recommendation system for car insurance," *European Actuarial Journal*, no. 10, pp. 377–398, 2020 [1].
- L. Lesage, M. Deaconu, A. Lejay, J. Meira, G. Nichil, and R. State, "Hawkes processes framework with a gamma density as excitation function: application to natural disasters for insurance," https://hal.inria.fr/hal-03040090, 2020 [2]. Accepted for publication on *Methodology and Computing For Applied Probability*.
- L. Lesage, M. Deaconu, A. Lejay, J. Meira, G. Nichil, and R. State, "A recommendation system for insurance built with a Multivariate Hawkes Process based on customers' life events," *https://hal.archives-ouvertes.fr/hal-03483812*, 2021 [3]. Submitted on *European Journal of Operational Research*.
- L. Lesage, "A Hawkes process to make aware people of the severity of COVID-19 outbreak: application to cases in France," *https://hal.archives-ouvertes.fr/hal-02510642*, 2020 [4].

Contents

List of Figures

List of Tables

List of Symbols and Acronyms

ALAE	Allocated Loss-Adjustment Expenses
GDPR	General Data Protection Regulation
IBCF	Item-Based Collaborative Filtering
MHP	Multivariate Hawkes Process
ROI	Return On Investment
SHAP	SHapley Additive exPlanation
STATEC	Institut national de la statistique et des études économiques du Grand-Duché de Luxembourg
SVD	Singular Value Decomposition
TF-IDF	Term Frequency–Inverse Document Frequency
UBCF	User-Based Collaborative Filtering

Chapter 0

Introduction

The main goal of the thesis is to build a recommendation system for Luxembourgish customers who subscribe to a car insurance. In this introduction, after describing the context of car insurance in Luxembourg, we present the motivation of the research. Classic approaches for recommendation systems could not fit to insurance context, so we propose a specific model based on a MHP (Multivariate Hawkes Process) fitted on customers' life events to achieve the main objective of the thesis. At the end of the introduction, we briefly sum up the thesis outline by listing all the chapters.

0.1 Background: car insurance in Luxembourg

Luxembourg: demographic context.
Luxembourg faced a strong demographic growth over the past twenty years. According to [5], there were 384k inhabitants in 2000 versus 626k in 2020, which represents a 63% growth explained mainly by population which doest not have Luxembourg nationality (161% growth). As a result, the growth of the total of registered vehicles in Luxembourg from 2000 to 2020 is of 60% as well, from 318k to 509k vehicles.

Car insurance in Luxembourg and Foyer Assurances.
Each and every vehicle from the Luxembourgian car fleet which is driven on public roads has to be insured and covered by the third party responsibility guarantee at least. Foyer Assurances is the leader of car insurance in Luxembourg, having a market share of 44% in non-life insurance. Foyer's car insurance product is characterized by a set of guarantees, with a particular structure. In this sense, the customers must select standard guarantees

(including third party liability), and could add optional guarantees. Given these guarantees, the customers could be covered for theft, fire, material damage, acts of nature, personal belongings, etc.

Existing commercial actions.
As a consequence, Foyer Assurances has an increasing number of car insurance policies every year, which implies more commercial animation and loyalty management. For now, agents are in charge of the sales initiatives, by contacting the customers in their portfolios whenever deemed relevant. Moreover, large scale marketing campaigns (e.g. a discount for subscribing a product) are performed by selecting customers randomly.

0.2 Research motivation: recommendation system

The objective is to build a recommendation system in order to help agents in their commercial animation by automatically selecting from their large portfolios the customers most likely to augment their insurance coverage. Thus, an insurance company using this solution could combine advantages from both data analysis and human expertise and optimize up-selling campaigns for instance. We insist on the fact that the goal is to support the agents that are and will continue to be the best advisers for customers, due to their experience and their knowledge of their portfolio. Agents validate if the recommendations from our system are appropriate to customers and make trustworthy commercial opportunities for them. The recommendation system is also planned to be integrated in customers' web-pages, in order to provide them a personalized assistance online. Indeed, more and more customers use the Foyer mobile app to manage their products and their claims.

We consider $\mathcal{U} = \{1, ..., U\}$ the set of customers who subscribed to an insurance product at Foyer. The objective of the recommendation system is to estimate, for each customer $u \in \mathcal{U}$, the probability that a customer would add an insurance cover at time t if recommended, denoted $p_u(t)$. This probability is built from all information available about the customer u that Foyer has collected until time t:

$$p_u(t) = \mathbb{P}\Big(\text{customer } u \text{ adds an insurance cover at time } t | \text{information about customer } u \text{ until } t\Big).$$

Thus, agents should address their recommendation to customers who have the highest probability to add an insurance cover in priority.

Main applications of recommendation systems.
Recommendation systems are currently adopted in many web applications. They offer a huge amount of products with daily use (e.g. e-commerce websites, music and video

streaming platforms), in order to make customer's decision-making easier and tackle problems related to over-choice. For most famous platforms, such as Amazon and Netflix, users must choose between hundreds or even thousands of products and tend to lose interest very quickly if they cannot make a decision (see [6]). Recommendation systems are then essential to give customers the best experience. In general, we can organize the recommendation systems in three types as follows.

First type of recommendation system.
Collaborative filtering is the first category of recommendation systems (see [7]). It consists in formulating recommendations by filtering information from many viewpoints or data sources. The first subset of collaborative filtering techniques is the so-called memory-based approach. This type of model compiles similarities and distances between users or items, from ratings given by users to items, or lists of items purchased by each user if there are no ratings. The idea is to identify for a user A either the most similar user B, then recommend to user A items that were already purchased by user B (User-Based Collaborative Filtering (UBCF), see [8]), or items that are the most similar to items user A has already subscribed to (Item-Based Collaborative Filtering (IBCF), see [9]). The second subset of collaborative filtering techniques is the so-called model-based approach, with data mining or machine learning algorithms. A classic model is based on matrix factorization, whose objective is to decompose the user-item interaction matrix (which contains ratings given by users to items), into the product of two matrices of lower dimensions. The main matrix factorization algorithm is Singular Value Decomposition (SVD) (see [10]).

Second type of recommendation system.
The second category of recommendation systems is the so-called content-based filtering. They analyze information about description of items and compile recommendations from this analysis. The main data source is text documents detailing content of items. A classic approach is Term Frequency–Inverse Document Frequency (Term Frequency–Inverse Document Frequency (TF-IDF), see [11]). Term Frequency counts the number of times a term occurs, while Inverse Document Frequency measures how rare a term is and how much information a term provides.

Third type of recommendation system.
The third category is the so-called hybrid filtering. It consists in mixing the two previous approaches and requires a huge amount of complex data. Deep learning techniques, which could be used for every type of recommendation system, are the most frequent approach to perform hybrid filtering. Given the tremendous improvement of computers' performances in the past few years, deep learning techniques deal with massive information and unstructured data. The survey in [12] lists the different deep learning techniques applied to recommendation systems, useful when dealing with sequences (e.g. language, audio,

video) and non-linear dependencies.

However, most of the algorithms we have described previously, which are appropriate for large-scale problems and to suggest the next best offer, would not fit for insurance covers recommendation. Indeed, the insurance context differs by three major particularities (see also [13]):

- **Data dimensions**: the number of covers is limited to a small number (e.g. 10-20) of guarantees. In comparison with thousands of books or movies proposed by online platforms, dimensions of the problem are reduced.
- **Trustworthiness**: insurance products are purchased differently from movies, books and other daily or weekly products. Frequency of contacts between an insurance company and customers is reduced since policyholders modify their cover rarely. Therefore, a high level of confidence in recommendations for insurance customers is needed. While recommending a wrong movie is not a big deal since the viewer will always find another option from thousands of videos, recommending an inappropriate insurance cover could damage significantly the trust of customers in their insurance company.
- **Constraints**: while any movie or any book could be enjoyed by anyone (except for age limit), several complex constraints exist when a customer chooses his cover. For instance, some guarantees could have an overlap, or some criterion linked to customers' profile (i.e. age limits, no-claims bonus level, vehicle characteristics, etc.).

That is why it seems necessary to develop a recommendation system specifically suited to our data. The architecture of the recommendation system should be designed with regards to these insurance data properties. The recommendation system must be accurate, but not only. The explainability of the model is very important, because a customer is more likely to accept a recommendation if he/she is told why this recommendation is suggested. Thus, the recommendation system should not be a "black-box" model. Moreover, the transparency of the model is necessary for the recommendation system to be in compliance with General Data Protection Regulation (GDPR), which requires accountability from Machine Learning decision-making systems.

0.3 Life events

Customers select their car insurance cover in function of several factors. Among all the features that should be included in the building of the recommendation system, we could mention:

- Their vehicle: a brand new and expensive car is more likely to have a full coverage than a second-hand vehicle, an electric car needs specific guarantees, etc.;
- Their income: customers could adapt their coverage to their financial capacity;
- Their location: several insurance guarantees are more suited to either urban areas or countryside (e.g. theft, acts of nature).

Not only the recommendation system must be accurate by suggesting an insurance cover suited to these customers characteristics, but also the timing of the recommendation is essential. Customers should be advised at the moment when they think that it is relevant for them to change their insurance cover. By observing the behaviour and the evolution of a customer in the insurance context, customers seem to modify their insurance cover when a significant event happens in their life. Among these life events, we could mention a birth, a marriage, a move, etc. Customers seem to think that one of these life events would be a good opportunity to reconsider their coverage. It is thus natural to propose and study how other life events could have an influence on customer choices on insurance cover. The recommendation system should update its output at any occurrence of a life event.

To the best of our knowledge, there exists no recommendation system in the literature dealing with life events. However, several recommendation systems are based on data about events in social networks. In [14], the so-called Outlife recommender suggests the most relevant event to attend with a group of Facebook friends. Authors in [15] compare several algorithms on Meetup data, a social network to organize online events between people who share similar interests. Other recommendation systems [16], [17] also deal with social events such as business or academic meetings, movie or restaurant nights, etc.

To sum up, the propensity for a customer to modify his insurance cover is first explained by a set of characteristics, which allow to evaluate an *a priori* result. We denote $\boldsymbol{x_u}(t)$ the vector of characteristics for customer u observed at time t. Moreover, the occurrence of life events should update the recommendations, which allows to offer an *a posteriori* point of view. This reasoning is similar to the calculation of insurance premiums in [18], based on an *a posteriori* vision which updates premiums as a function of the occurrence of claims. We denote $s_u(t)$ the sequence of events that occurred until time t. We assume that the two visions are independent, since $\boldsymbol{x_u}(t)$ contains no information about life events.

Assumption 1 *The model of the recommendation system is based on a learning which excludes data about life events. The final model is the sum of this learning output and a term which only depends on life events occurrences.*

Thus, we model $p_u(t)$ by introducing functions f_1 and f_2 such that:

$$p_u(t) = \underbrace{f_1(\boldsymbol{x_u}(t))}_{\substack{\text{a priori vision} \\ \perp \text{ life events}}} + \underbrace{f_2(s_u(t))}_{\substack{\text{a posteriori vision} \\ \text{depends on life events}}} ,$$

where the *a priori* and *a posteriori* visions are considered independent and where functions f_1 and f_2 must be estimated. In compliance with this modelling, we propose to build the recommendation system thanks to a Multivariate Hawkes Process, which is a category of point processes.

0.4 Modelling: Multivariate Hawkes Processes

Point processes are particularly suited to model the occurrence of random events in time. Several works in the literature use point processes to model real-world event sequences [19], such as patient flows in a hospital (transition of patients among care units) [20], conflicts (Afghan war) [21], earthquakes [22] or queuing (arrival of customers in a queue) [23].

A point process is a sequence of event times $\{t_1, t_2, ..., t_n, ...\}$ (see Section 2.2.1). We associate a counting process $N(t)$, which represents the number of events that occurred until time t. Point processes are characterized by their conditional intensity function [24], also called hazard function, denoted $\lambda^*(\cdot)$. We define $(\mathcal{H}(t), t \geq 0)$ the filtration of the history of data about customers until time t.

Definition 1 ***(Conditional intensity function: expected rate of occurrences conditioned on $\mathcal{H}(t-)$).*** *Let us consider a counting process $N(\cdot)$. If $\lambda(t|\mathcal{H}(t-))$ exists such that:*

$$\lambda(t|\mathcal{H}(t-)) = \lim_{h \to 0^+} \frac{\mathbb{E}[N(t+h) - N(t)|\mathcal{H}(t-)]}{h}, \tag{1}$$

then $\lambda(t|\mathcal{H}(t-))$ is the conditional intensity function of $N(\cdot)$. We denote $\lambda(t|\mathcal{H}(t-))$ by $\lambda^*(t)$.

If we consider an infinitesimal time interval dt, the probability that an event occurs between t and $t + dt$ is proportional to dt and $\lambda^*(t)$:

$$\mathbb{P}\Big(N(t+dt) - N(t) = 1 \Big| t_1, t_2, ..., t_{N(t)}\Big) = \lambda^*(t)dt.$$

For each customer, we model the event "Subscription to a new insurance cover" as a point process and $\lambda_u^*(t)$ represents the propensity for the customer u to follow the recommendation at time t. We associate the counting process $N_u(t)$, representing the number of times the customer u has modified his insurance cover until time t. Therefore, the higher $\lambda_u^*(t)$ is, the more the customer is likely to accept the suggestion from the recommendation system at time t.

Considering the modification of the insurance cover as a point process, $p_u(t)$ is proportional to the hazard function and that is why the recommendation system could be based on $\lambda_u^*(t)$. We consider now the following model:

$$\lambda_u^*(t) = \underbrace{\lambda_1(\boldsymbol{x_u}(t))}_{\text{a priori vision}} + \underbrace{\lambda_2(s_u(t))}_{\text{a posteriori vision}} \quad , \tag{2}$$

where λ_1 and λ_2 are now the functions to estimate.

If the conditional intensity function of a point process is kept fixed, it corresponds to a homogeneous Poisson process. If it depends on the time but not on past events, it corresponds to a non-homogeneous Poisson process. For our purpose, the conditional intensity function modelling the subscription of a new insurance cover should depend on customer's characteristics (*a priori* vision) and on the history of several types of events: life events and past subscriptions themselves (*a posteriori* vision).

That is why we select a Multivariate Hawkes Process to model the recommendation system. A Multivariate Hawkes Process is a set of several point processes, where each process corresponds to a type of event. These point processes are not independent: each process is influenced by the past occurrences of every type of event, including himself. It is the so-called mutually exciting effect. Therefore, we construct a Multivariate Hawkes Process where one process is the event "Subscription to a new insurance cover" described above and the other processes are the life events deemed relevant to this study (e.g. marriage, birth, move, etc.).

Definition 2 ***(Multivariate Hawkes processes).*** *We consider a customer u. For each type of event $i \in \{1, ..., m\}$, we consider:*

- $\lambda_i(\boldsymbol{x_u}(t)) > 0$ *the background intensity (we denote $\boldsymbol{\lambda}(\boldsymbol{x_u}(\boldsymbol{t})) = \left[\lambda_i(\boldsymbol{x_u}(t))\right]_{i\in\{1,\dots,m\}}$);*
- $\mu_{i,j} : [0, +\infty[\to [0, +\infty[$, $j \in \{1, ..., m\}$, *the triggering function that models the effect of event j on event i;*
- $s_u(t) = \{(e_k^u, t_k^u)_{k=1}^{N^u(t)}\}$ *the sequence of events observed for customer u until time t: e_k^u is the number of the k^{th} event and t_k^u the corresponding time. $N^u(t)$ is the total number of events that occurred to u*

The set of counting processes $\{N_1^u(\cdot), ..., N_m^u(\cdot)\}$ is a multivariate Hawkes process if the conditional intensity function of each process $N_i^u(\cdot)$ is of the form:

$$\lambda_i^{u,*}(t) = \lambda_i(\boldsymbol{x_u}(t)) + \sum_{j=1}^{m} \int_0^t \mu_{i,j}(t-v)dN_j^u(v) = \lambda_i(\boldsymbol{x_u}(t)) + \sum_{k \in s_u(t)} \mu_{e_k^u}(t - t_k^u). \quad (3)$$

Each time a life event occurs, it affects the results from the recommendation system. For each process, the conditional intensity function is the sum of two terms: the so-called background intensity, which does not depend on events history but should depend on customers' characteristics in the recommendation system to perform personalized suggestions (*a priori* vision), and the sum of influences of past events modelled by the so-called triggering functions (*a posteriori* vision):

$$\begin{aligned} \lambda_1^{u,*}(t) &= \underbrace{\text{Background intensity}(\boldsymbol{x_u}(t))}_{\text{a priori vision}} + \underbrace{\text{Sum of mutually-exciting effects}(s_u(t))}_{\text{a posteriori vision}} \\ &= \lambda_1(\boldsymbol{x_u}(t)) + \sum_{k \in s_u(t)} \mu_{e_k^u}(t - t_k^u), \end{aligned}$$

where each life event k that occurred before t influences $\lambda_u^*(t)$ by summing $\mu_{e_k^u}(t - t_k^u)$ (see Definition 3). Thus, the objective is to estimate the background intensity function λ_1 and all the triggering functions, so that given the characteristics of each customer u and the history of their life events the recommendation system estimates $\lambda_u^*(t)$. Therefore, the customers with the highest value of hazard function are most likely to increase their insurance cover and should be targeted by agents in priority.

Several models based on Hawkes Processes were developed to build a recommendation system. The RecSys Challenge 2016 [25] aimed to create a recommendation system to suggest the most suited job to users: authors in [25] proposed an approach with a Hawkes Process. Temporal user-item interactions are modelled with a Multivariate Hawkes Process in [26] and are applied to Internet Protocol television (IPTV), Yelp and Reddit data. In [27], authors construct a recommendation system on music songs and shopping items in order to estimate not only the right moment to make a recommendation, but also to predict the next returning time to the item. A point of interest recommendation system based on geographic spatial data is also proposed in [28].

All these approaches are powerful and very interesting but not useful for our problem as most of these papers use data quite different from our purpose, in terms of dimensions, frequency, etc. The recommendation system proposes several contributions. First, for most of approaches presented in the related works, estimated background intensities are identical for every user (i.e. background intensity does not depend on customers' characteristics: $\lambda_1(\boldsymbol{x_u}(t))$ does not depend on $\boldsymbol{x_u}(t)$). Here, we should propose to integrate a personalized background intensity for each customer, based on either Machine Learning or relevant statistical analysis. Moreover, we should propose triggering functions which are adapted to the insurance data to our disposal and which are different from classic kernels. Also, parameters estimation should be suited to our data. Working on real data from customers could imply a lack of data quality, especially when using a temporal point process approach which requires a reliable report of events' dates of occurrence. That is why the objective function based on maximum of likelihood should be penalized in a new way, in order to improve the estimated Hawkes Process robustness and compensate several imperfections on real data.

0.5 Implementation: R package

The recommendation system building introduces specific methods in terms of estimation, penalization, simulation, data-visualization, etc. for MHP. It implies the implementation of these new tools, that we gather into a new R package called `[libraryname]`.

For now there exists no R packages which implement functions to estimate, plot or simulate MHP by letting the choice of the triggering functions in parameters, necessary for the implementation of the recommendation system. Existing R packages, such as `hawkes` [29], `emhawkes` [30], `hawkesbow` [31] or `PtProcess` [32] include one or several functionalities but none of them takes into account all the singularities of the recommendation system. It is also the case for Python packages such as `pyhawkes` [33] or `tick` [34].

`[libraryname]` provides to the R community a package to build a recommendation system based on a MHP and a wide range of plotting tools around MHP. It proposes an implementation of the parameters estimation algorithm presented in Section 2.4.2, of the Ogata's thinning algorithm (see Section 2.2.5.2) and a computation of MHP moments (see Section 3.3.2.2). Several plot functions are included, in order to clearly display as much information as possible about MHP. The `[libraryname]` package provides also examples and a full documentation.

0.6 Contributions

To sum up, the purpose of the thesis is to build a recommendation system, in order to help agents selecting the customers who are most likely to increase their insurance cover. The main contributions are to:

1. Build a Multivariate Hawkes Process based on life events which models the probability to add an insurance cover, by estimating the background intensity function by a Machine Learning approach and the triggering functions introduced previously;

2. Provide explainability to the model so that customers know why an additional insurance cover is recommended;

3. Propose an implementation of all the computations linked to the recommendation system (i.e. estimation, data-visualization, etc.) into a R package.

0.7 Thesis outline

The thesis is presented in five chapters. Chapter 1 explains how the background intensity of the Multivariate Hawkes Process is computed thanks to a Machine Learning algorithm, so that each customer has a personalized recommendation. Chapter 1 is shown an extended version of the method presented in [1], in which the method is used to make the algorithm explainable. Chapter 2 presents a Multivariate Hawkes Processes framework in order to compute the dependency between the propensity to accept a recommendation and the occurrence of life events (inspired from [2] and [3]): definitions, notations, simulation, estimation, properties, etc. Chapter 3 presents several results of the recommendation system: estimated parameters of the model, effects of contributions, back-testing of the model's accuracy, etc. Chapter 4 presents the implementation of our work into a R package. Chapter 5 concludes the thesis by listing the main contributions and proposing several work perspectives and improvements about the current recommendation system.

Chapter 1

Influence of customer's profile: Machine Learning and explainability

1.1 Objective

In Assumption 1 and Equation (2), we introduced the main assumption of the model developed for the recommendation system. We assume that the model is learned on two sources of data, in an independent way. The first source is the set of data available for customers which excludes life events information, the second source is the sequence of life events that occurred in customer's history (see Chapter 2).

The learning based on customers' data excluding life events is detailed on this chapter. The objective is to provide a baseline of the propensity for a customer to accept a recommendation (i.e. the *a priori* vision described in the Introduction), based on their background information: personal records, location, income, current insurance cover, etc. In other words: how should the recommendation system be built without any information about life events? The model is the first step of the MHP building since it will be used as the background intensity for the point process whose conditional intensity function provides the output of the recommendation system.

In order to provide an effective way to predict an output from a huge amount of data about customers, a Machine Learning approach was taken on. The objectives are to select the most accurate algorithm and to provide explainability on predictions, so that customers could be aware of the reason why they are advised to augment their insurance cover.

Chapter 1 is organized as follows. Section 1.2 introduces the target variable which is learned by the algorithm. Section 1.3 describes the data preparation before learning. Section 1.4 formulates the problem to be solved. The algorithm selected is described in Section 1.5, while Section 1.6 provides a method to make this algorithm explainable. Section 1.7 presents the results of a pilot phase where the approach was tested on real customers.

1.2 Target variable

Tom Mitchell defines Machine Learning as follows (see [35]): *"Machine Learning is the study of computer algorithms that improve automatically through experience. A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*. The task to learn T is to predict the customers who are the most likely to accept a recommendation for an additional insurance cover. The problem is to find the algorithm E which learns T the best according to a performance measure P that must be defined. For this task T, the algorithm E should belong to one specific category: supervised learning algorithms.

According to [36], *"supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs."*. It consists in learning a target variable from training data which are labelled by this target variable. For the task T, training data is a set of information collected about a sample of customers. Which target variable should be considered here?

Ideally, since the objective is to learn whether a customer would accept a recommendation or not, training data should be labelled with information about observed recommendations (e.g. a binary variable which indicates whether a customer has accepted a recommendation given his profile). Problem is that there is no historical data on such previous recommendation campaign: it is the first time that Foyer tries this kind of automatized up-selling method. It corresponds to the so-called cold-start problem for recommendation systems (described in [37]), when there is a lack of observed interactions between users (i.e. the customers) and items (i.e. the insurance covers).

In a first approach, the learning process is made on customers who added a guarantee either by themselves or by implicit recommendations from prior contacts between agents and customers, and not necessarily on customers who improved their cover as a direct consequence of an explicit recommendation. The target variable is binary and depends on observation time. We denote $y_u(t)$ the value of the target variable for customer u observed at time t. For each customer u, if u has added an insurance cover between times t and $t+1$, then $y_u(t) = 1$, else $y_u(t) = 0$.

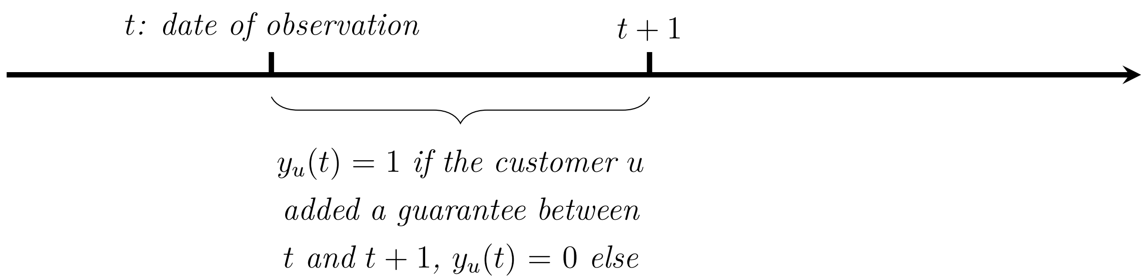
Notation 1 *We consider:*

1. $\mathcal{U} = \{1, ..., U\}$ *the set of customers, where* U *is the total number of customers;*

2. $\boldsymbol{y}(t)$ *the target variable, calculated at time* t*:*

$$\boldsymbol{y}(t) = (y_u(t))_{u \in \mathcal{U}}, \tag{1.1}$$

where $y_u(t) \in \{0, 1\}$ *is the label value for the customer* u*. The scheme below illustrates the construction of* $\boldsymbol{y}(t)$*.*



This approach seems to be the best alternative. Even if there is *a priori* no added value to suggest a guarantee to a customer who could have added the guarantee voluntarily, this method allows us to learn the profile of the customers who would be less reluctant to add a guarantee: we believe that a customer who is very hesitant to improve his cover by himself would not listen to his agent if he suggested him to do so. But we should keep in mind that the crucial point of a recommendation is the receptiveness of a customer to his agent and his propositions, which is hard to evaluate without data about human interactions with the customer.

1.3 Data preparation

In a Machine Learning project, most of time is dedicated to data preparation. In [38], the proportion of time necessary is evaluated to 80%. Once the problem is defined, data preparation is the step which turns all the data collected about customers into a dataset that we can use for modelling. According to [39], data preparation refers to the addition, deletion, or transformation of training data.

Notation 2 *We consider:*

1. $\boldsymbol{x_u}(t)$ *the vector of characteristics describing the profile of the customer* $u \in \mathcal{U}$*, excluding information about life events, observed at time* t*;*

2. *$\boldsymbol{x}(t)$ the dataset gathering information about all customers in $\mathcal{U}$:*

$$\boldsymbol{x}(t) = (\boldsymbol{x}_{\boldsymbol{u}}(t))_{u \in \mathcal{U}}; \tag{1.2}$$

3. *F the number of features which describe any customer u, i.e. the number of columns of $\boldsymbol{x}(t)$;*

4. *$\mathcal{F}$ the set of the features.*

The dataset should contain as much relevant information about customers which could influence the fact that they modify or not their insurance cover as possible. As a reminder, information about life events are excluded here. The dataset $\boldsymbol{x}(t)$ is built from multiple data sources, gathered in an unique table. The main categories of data contained in $\boldsymbol{x}(t)$ are listed below.

- Current car insurance coverage: in order to determine whether a customer should get an additional cover, the dataset should include information about current coverage. Data describe the guarantees already subscribed, insured vehicle's characteristics (e.g. price, model, engine power), premium amounts, no-claims bonus scale level, etc.;

- Other insurance products subscribed: if a customer got a complete cover on another insurance product (e.g. home, health, pension, savings), he may be more likely to accept another guarantee for his car insurance. Data include information such as the number of other insurance products the customer has subscribed to and the completeness level of the coverage;

- Contacts between customers and Foyer: the history of interactions between customers and their insurance company could give an insight about how they would react to a recommendation made by their agent. Data give information about the quantity and the reasons of phone calls, mails or mobile app views;

- Claims rate: if a customer already had an accident, especially an accident not covered by his current coverage, he may accept an additional guarantee to be covered the next time. Data give information about the number, the type and the location of past accidents;

- Personal records: several personal information could make a difference on the selection of insurance cover. Data include age, address, information about family, home, Foyer agency, date of the first subscription, etc.;

- Open data: all the previous data are from Foyer databases. Open data enriches the dataset with information such as demography, crime rate, etc. that could be joined to the age or location of customers. The main source of these public data is the Luxembourgish institute of statistics: Institut national de la statistique et des études économiques du Grand-Duché de Luxembourg (STATEC).

For the purpose of this recommendation system, the algorithms tested to learn the target variable expect data in a tabular structure with numerical values, i.e. $\boldsymbol{x}(t) \in \mathbb{R}^{U \times F}$. Therefore, some data would need transformations to be usable for algorithms. Feature engineering [40] allows us to build relevant features based on existing variables from raw data. It could be an aggregation of several features, or a transformation from categorical to numeric feature. This step is in general based on knowledge of datasets and on intuition supported by experts from specific fields about what could be the most explanatory features.

Example 1 *Addresses as character strings were transformed into GPS coordinates so that they are in a numeric format. Moreover, a feature which equals the distance between customer's home and his insurance agency was calculated from these coordinates.*

Therefore $\boldsymbol{x}(t)$ is of dimensions U rows (i.e. customers) and F columns (i.e. features). We also mention that there exists automated methods to perform feature engineering such as deep feature synthesis (see [41]), but results were inconclusive in our study, due to the specificity of used datasets which required to aggregate them manually.

1.4 Problem definition

We consider the couple of random variables $(\boldsymbol{X}, Y)$, where each couple $(\boldsymbol{x_u}(t), y_u(t))$ is an observation of these random variables. The objective is to estimate the function f such that $\mathbb{E}(Y|X) = f(X)$. Since $Y \in \{0, 1\}$, $f(X) = \mathbb{P}(Y = 1|X)$.

This function is learned from the set of observations $\{(\boldsymbol{x_u}(t), y_u(t)), u \in \mathcal{U}\}$ by several potential Machine Learning algorithms categories, among the following list:

- Linear regression;
- Decision tree;
- Random Forest;
- SVM;
- Gradient Boosting.

We select the algorithm which has the best performance according to the so-called k-fold cross validation method (see [42]), presented in Algorithm 1. It consists in dividing the dataset of observations in k random samples. Then, for each group $i \in \{1, ..., k\}$, training the algorithm on the other $k-1$ groups and measuring the error of the algorithm on the

i^{th} group. We select empirically $k = 10$. The selected algorithm is the one with the lowest mean error.

Algorithm output is $\hat{f}$, estimation of f. Therefore, the prediction of the target variable for any customer u is $\hat{y}_u(t) = \hat{f}(x_u(t))$. To calculate the error of the algorithm, we introduce a loss function denoted Ψ. Over a set of customers $\mathcal{U}_s \subseteq \mathcal{U}$, the error equals:

$$\text{Error}(\mathcal{U}_s) = \frac{1}{\#\mathcal{U}_s} \sum_{u \in \mathcal{U}_s} \Psi(y_u(t) - \hat{y}_u(t)), \tag{1.3}$$

where $\#\mathcal{U}_s$ denotes the cardinal number of $\mathcal{U}_s$.

Algorithm 1: k-fold cross validation

Inputs: dataset $\boldsymbol{x}(t)$, target variable $y(t)$

Result: Algorithm error

Divide dataset $\boldsymbol{x}(t)$ in k random samples;

for $i \in \{1, ..., k\}$ **do**

  Learn f using data from all groups but the i^{th};

  Predict the target variable for customers of group i;

end

Calculate error over all predictions.

1.5 Algorithm selection

1.5.1 Algorithm: XGBoost

After performing k-fold cross validation, the XGBoost algorithm was selected, based on Gradient Boosting method. Gradient Boosting is a sequential ensemble method, first proposed by Breiman and developed by Friedman in [43]. The principle of boosting is to combine weak learners (e.g. decision trees for Gradient Tree Boosting) trained in sequence to build a strong learner. In Gradient Tree Boosting, each decision tree attempts to correct errors made by the previous tree. At each iteration, a decision tree is fitted to residual error. Algorithm 2 presents the generic Gradient Tree Boosting method.

Algorithm 2: Friedman *Gradient Tree Boosting*

Inputs: training dataset $\boldsymbol{xs}(t) = (\boldsymbol{x}_u(t))_{u\in\mathcal{U}_s}$, target feature $\boldsymbol{y}(t) = (y_u(t))_{u\in\mathcal{U}}$, loss function Ψ, number of iterations B

Result: Vector $\hat{\boldsymbol{y}}(t)$, estimation of $\boldsymbol{y}(t)$, the probability of adding a guarantee

Initialize $\hat{\boldsymbol{y}}(t)$: $\forall u \in \mathcal{U}_s$, $\hat{y}_u^{(0)}(t) = \underset{\rho}{\operatorname{argmin}} \sum_{u\in\mathcal{U}_s} \Psi(y_u(t), \rho)$;

for $m \in \{1, ..., B\}$ **do**

Compute negative gradients: $z_u = -\frac{\partial \Psi[y_u(t), \hat{y}_u(t)]}{\partial \hat{y}_u(t)}\Big|_{\hat{y}_u(t)=\hat{y}_u^{(m-1)}(t)}$, $u \in \mathcal{U}_s$;

Train a decision tree h using the dataset $\{\boldsymbol{x}_u(t), z_u\}_{u\in\mathcal{U}_s}$;

Compute step size: $\rho \leftarrow \underset{\rho}{\operatorname{argmin}} \sum_{u\in\mathcal{U}_s} \Psi[y_u(t), \hat{y}_u^{(m-1)}(t) + \rho \times h(\boldsymbol{x}_u(t))]$;

$\hat{y}_u^{(m)}(t) \leftarrow \hat{y}_u^{(m-1)}(t) + \rho \times h(\boldsymbol{x}_u(t))$, $u \in \mathcal{U}_s$.

end

XGBoost implementation is characterized by:

- Parallel learning: XGBoost uses multiple CPU cores to perform parallelization to build decision trees and reduces computation time,
- Regularization: XGBoost adds a regularization term which avoids over-fitting and then optimizes computation.

The other algorithms tested are:

- A single CART decision tree (see [44]);
- Random Forest (see [45]), ensemble learning method which applies bootstrap aggregating to decision trees.

1.5.2 Hyperparameters

The hyperparameters introduced previously are as follows:

- the portfolio contains $U = 57.000$ policyholders of the car insurance product;
- there are $F = 165$ features in the dataset $\boldsymbol{x}(t)$;
- the loss function Ψ is the mean square loss function: $\Psi(x) = \frac{x^2}{2}$. For a regression problem, the two common choices are the mean square and the mean absolute loss functions. The first one penalizes the model for huge errors by squaring them, but it is less robust to outliers than the second one as a consequence [39];
- XGBoost uses $B = 1.000$ iterations;
- Time is fixed at $t =$ December 2017, so that the target variable is based on insurance cover modifications observed during year 2018: for each customer $u \in \mathcal{U}$, $y_u(t) = 1$ if and only if the customer u has added a cover during 2018. Over all the customers, we observe that $\frac{1}{U}\sum_{u=1}^{U} y_u(t) = 12\%$.

1.5.3 Back-testing

We perform a back-testing of this algorithm. The learning is based on what happened during year 2018: the objective is to test on year 2019 if, in the case we would have recommended guarantees based on our method, customers would have followed the recommendations.

Since the objective is to avoid wrong suggestions as much as possible, at the risk of limiting the amount of recommendations, we evaluate this step on the customers that are most likely to accept an additional cover. To do so, we plot in Figure 1.1 the rate of customers sorted by decreasing probability of acceptance who added at least one guarantee in the past (y-axis) versus the top $x\%$ of customers sorted by probability of acceptance (x-axis). The reference curve (in red) is the result given by a perfect model, which would rank every addition of cover on highest probabilities. We compare the model with other methods of supervised learning tested.

Let us further discuss an example. The point highlighted in purple says that, if we consider the 10% with the highest probabilities of adding a guarantee according to the XGBoost algorithm, 66% of these 10% indeed added a guarantee in the past.

Figure 1.1 shows that the XGBoost algorithm is the most accurate method, since XGBoost is the closest curve to the reference model on a major part of the top 20% of customers.

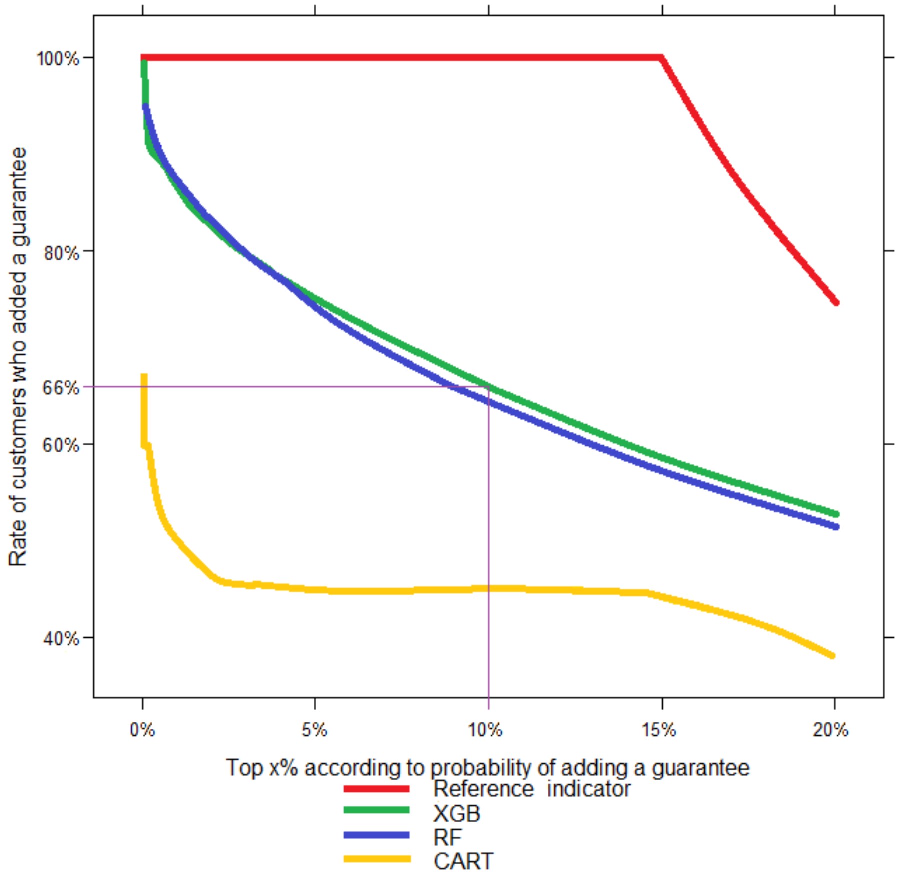


Figure 1.1: Back-testing of XGBoost algorithm, on every customer

1.6 Explainability

In this section, the objective is to propose a method which allows us to add explainability to the model described previously. It is very useful for both agents and customers to know why a recommendation is made: the agent could validate whether the suggestion makes sense or not and the customer is aware of the reasons why he should increase his cover. More precisely, for each customer u, the model outputs a probability to add an additional guarantee: the objective is to evaluate the contribution of each feature used for training in the calculation of the probability.

1.6.1 SHAP values

The contribution of features for each prediction is evaluated by their SHapley Additive exPlanation (SHAP) values, first introduced in [46]. SHAP values are inspired from game theory, and more precisely from Shapley values, defined in [47]. Considering a game where several players collaborate to obtain an overall gain, Shapley values quantify the contribution of each player.

We consider S a subset among a set of players $\mathcal{N}$ (which contains M players) and we introduce v the characteristic function. The quantity $v(S)$ represents the expected gain players from S could obtain if they cooperate. The Shapley value of player $i \in \mathcal{N}$, denoted by $\phi(i)$, equals:

$$\phi_i = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} (v(S \cup \{i\}) - v(S)). \tag{1.4}$$

SHAP values are a specific type of Shapley values. The features used to learn a model are considered as the players, whose collaboration leads to the overall gain, i.e. the prediction of the target variable. Each feature has a different contribution with regard to the customer.

Notation 3 *We denote:*

1. *$\phi_i^u(t)$ the contribution of feature $i \in \mathcal{F}$ to the prediction for the customer $u \in \mathcal{U}$ observed at time t;*
2. *$S \subseteq \mathcal{F}$ a subset of features;*
3. *$\boldsymbol{x}_u^S(t)$ the values of features from subset S for the customer u observed at time t.*

In [46], the characteristic function for SHAP values is defined as follows:

$$v_t^u(S) = \mathbb{E}[f(X_S, X_{\bar{S}}) | X_S = \boldsymbol{x}_u^S(t)] = \int f(x_S, x_{\bar{S}}) \mathbb{P}(x_{\bar{S}} | x_S = \boldsymbol{x}_u^S(t)), \tag{1.5}$$

which corresponds to the expected output of the model, conditional on the values of the subset of features S, and where the union of X_S and $X_{\bar{S}}$ equals X as defined in Section 1.4. Therefore the contribution of the feature $i \in \mathcal{F}$ for the customer $u \in \mathcal{U}$ equals:

$$\phi_i^u(t) = \sum_{S \subseteq \mathcal{F} \setminus \{i\}} \frac{|S|!(F - |S| - 1)!}{F!} (v_t^u(S \cup \{i\}) - v_t^u(S)), \tag{1.6}$$

where F is the total number of features, introduced in Notation 2.

1.6.2 Computation: dependencies between features

The computation of SHAP values is challenging because it requires the conditional distributions $\mathbb{P}(x_{\bar{S}} | x_S)$. If we assume independence between all the features, the Kernel SHAP method presented in [46] proposes an approximation of the computation of v. In [48], three types of dependencies between features are taken into account:

- Gaussian distribution: we assume that all the features follow a multivariate Gaussian distribution;
- Gaussian copula: the marginals follow their empirical distribution, but the global structure of the dependency between features is a Gaussian copula;

- Non-parametric estimation: authors in [48] propose to build an empirical conditional distribution based on the Mahalanobis distance (see [49]) between training instances.

Concerning the dataset $\boldsymbol{x}(t)$, we assume that the features are independent, except for one couple of variables: claims payments and Allocated Loss-Adjustment Expenses (ALAE). ALAE are the costs linked to the investigation an insurance company settles when a claim occurs, so that they pay the right amount of money to the customer. In [50], the authors model the structure of the dependency between the couple of variables based on real insurance data from [51] by testing several categories of copulas. In [52], the authors fit an Archimedian copula to loss-ALAE data. The Kendall tau (see [53]) of the two variables which measures the rank correlation equals 0.48, which could indicate that a copula is appropriate to fit the data.

In the bivariate case, a copula could be defined as follows.

Definition 3 *A copula C is a multivariate cumulative distribution function on $[0,1]^2$, where both marginal probability distributions are uniform.*

The main result about copulas is Sklar's theorem [54], written for the bivariate case in Theorem 1 below.

Theorem 1 *For each couple of random variables (X_1, X_2) of cumulative distribution function F with marginals F_1 and F_2, there exists a copula C such that $F(x_1, x_2) = C(F_1(x_1), F_2(x_2))$. The converse is true: given a copula C and marginals F_1 and F_2, $C(F_1(x_1), F_2(x_2))$ defines a cumulative distribution function.*

To establish the dependency between the two features, the type of copula is selected thanks to the *VineCopula* package [55], which estimates the parameters of a list of copulas by maximum likelihood estimation and evaluates the goodness-of-fit by a Kendall's process (see [56]). Table 1.1 presents the results of the comparison between the different types of copulas.

Table 1.1: Modelling of claims payments/ALAE data: comparison of several types of copulas

Copula type	p-value	AIC
Gaussian	0	-15.266
Clayton	0	-14.326
Gumbel	0.01	-18.108
BB1	0.15	-18.912

According to this test, the best fit is the so-called survival BB1 copula [57], which is defined by [58] as:

$$C(u_1, u_2, \theta, \delta) = [1 + [(u_1^{-\theta} - 1)^{\delta} + (u_2^{-\theta} - 1)^{\delta}]^{\frac{1}{\delta}}]^{-\frac{1}{\theta}}, \theta > 0, \delta \geq 1. \tag{1.7}$$

The parameters estimation by *VineCopula* package leads to $\theta = 0.33$ and $\delta = 1.64$. Figure 1.2 plots 15,000 simulations of the estimated copula.

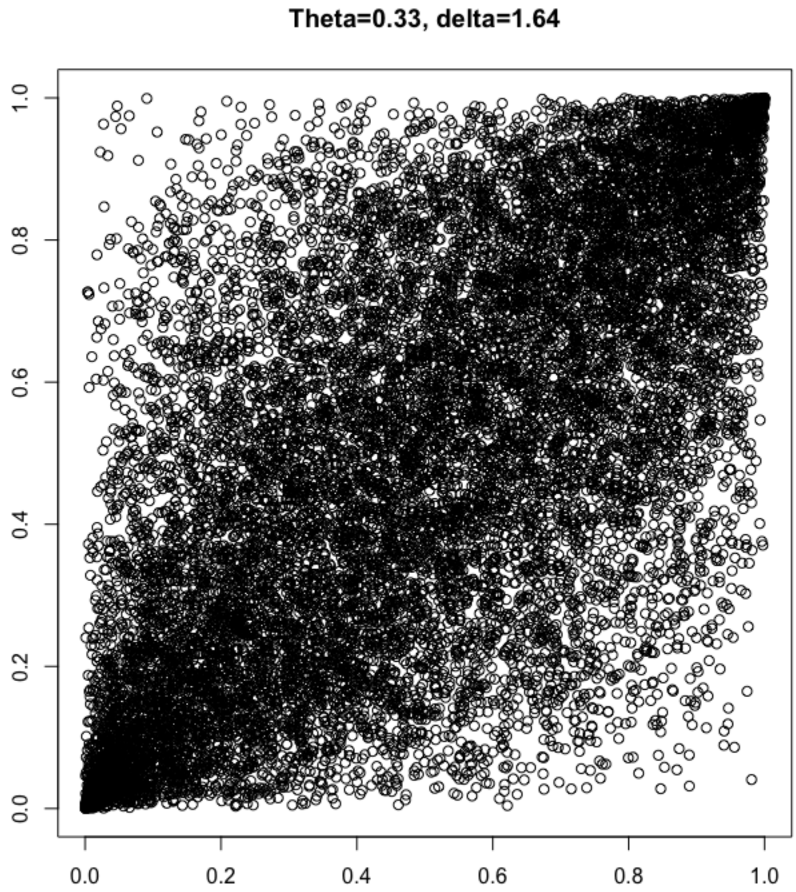


Figure 1.2: 15.000 simulations of a BB1 copula with parameters $\theta = 0.33$ and $\delta = 1.64$

The set of SHAP values for each customer provides an interpretation of each prediction as follows:

$$y_u(t) = \phi_0 + \sum_{i \in \mathcal{F}} \phi_i^u(t), \tag{1.8}$$

where ϕ_0 is the average value of the target variable over every customer and $y_u(t)$ is defined in 1.1. Therefore, the probability to add an insurance cover is the sum of the average probability and the contributions of features through the SHAP values.

1.6.3 Results

The implementation of SHAP values is inspired from *shapr* package [59], which proposes the three dependency structures described in [48] and listed previously. There exists several other R packages implementing SHAP values, among the 27 libraries listed in the survey [60]. Among a list of 27 R packages, the survey [60] identifies several other libraries which implement SHAP values as well. The package [59] was selected because we could adapt the sampling functions so that the two features following a copula structure could be sampled according to a BB1 copula, and the others variables in a independent way.

Since the decomposition from Equation (1.8) is unique for each customer, we propose several aggregated results. In particular, we focus on customers with the highest and lowest probability to augment their cover in order to answer the following question: which features make customers the most/less privileged target for the recommendation system? Table 1.2 presents quantiles of SHAP values from the most influent variables for the customers in the top 10% of prediction values. The overall average of predictions ϕ_0 equals 0.15 and the average of these top 10% customers is 0.65.

Table 1.2: Quantiles of SHAP values observed for the 10% customers with the highest probability to add an insurance cover

Feature	Min	25% quant.	Median	Mean	75% quant.	Max
Number of contacts with Foyer	−0.14	0.16	0.21	0.20	0.25	0.40
Number of guarantees subscribed	−0.04	0.01	0.06	0.07	0.09	0.12
Number of claims non covered	−0.02	−0.00	0.00	0.02	0.00	0.30

Table 1.2 illustrates that each feature influences predictions in a different way. The distribution of the number of contacts between Foyer and the customer shows that most of customers with a high probability are strongly influenced by this feature. On the other way, several features have a global neutral impact but influence deeply the prediction on a few customers. For instance, the influence of the number of claims non covered is weak for most of customers but could increase significantly the prediction for a few cases. Table 1.3 proposes the same analysis for the 10% customers with the lowest probability to augment their insurance cover.

Table 1.3: Quantiles of SHAP values observed for the 10% customers with the lowest probability to add an insurance cover

Feature	Min	25% quant.	Median	Mean	75% quant.	Max
Number of contacts with Foyer	−0.14	−0.08	−0.07	−0.07	−0.06	0.07
Number of products subscribed	−0.08	−0.03	−0.02	−0.02	−0.02	−0.01
Age	−0.21	−0.01	0.00	−0.03	0.01	0.01

In a similar fashion, most of customers with a low probability receive a negative influence from the feature representing the contacts between Foyer and the customer. Age has a weak impact on more than 75% of the customers but could lead to a low probability for a few people.

1.7 Tests on real customers

1.7.1 Context

In order to validate the results obtained by this method and before including these results into the model of the final recommendation system, we perform a pilot phase on real customers in which four Foyer agencies recommend an additional insurance cover to their clients, based on the output of the XGBoost algorithm. Among the portfolio of these agencies, we extracted the customers in the top 10% of the estimated probabilities of accepting a recommendation. Then the four agents proposed a recommendation to this selection of around 150 customers, by mail, phone or an appointment.

The four agents who took part in the pilot phase were selected thanks to a large portfolio and a strong motivation to test this experimental approach. Before the campaign, a presentation allowed them to discover how the recommendation system works. During the campaign, the recommendations were transmitted through a software which is daily used by agents to manage their commercial opportunities. After a recommendation, the agent typed in this interface whether the customer accepted or not, which allowed us to get the information very easily and continuously. After the campaign, agents shared their feedback during a meeting and suggested relevant potential improvements.

During the pilot phase, the agents propose to customers selected by the algorithm described previously to augment their car insurance cover. In practice, they recommend a guarantee among those the customer has not subscribed yet. Therefore, this test version of the recommendation system should propose a way to select the most appropriated guarantee for each customer.

An interface was developed in order to manage the pilot phase. We refer to Appendix A for a presentation of this tool.

1.7.2 Structure of the car insurance product

We describe here the structure of the car insurance product, in order to introduce the necessary notations to construct the algorithm which associates to each customer the most appropriated guarantee.

Notation 4 *We consider:*

1. *M the number of guarantees available for the car insurance product. M_s is the total number of standard guarantees and M_o the total number of optional guarantees. This leads to:*
$$M = M_s + M_o; \tag{1.9}$$

2. *$\mathcal{U}$ the set of the U customers who subscribed to the car insurance product;*

3. *$\mathcal{G}$ the set of M available guarantees:*

$$\mathcal{G} = \{g_1, ..., g_M\}, \tag{1.10}$$

where $g_1, ..., g_M$ are the guarantees. $\mathcal{G}$ is split into two disjoint subsets, $\mathcal{G}_s$ the subset of standard guarantees and $\mathcal{G}_o$ the subset of optional guarantees:

$$\mathcal{G} = \mathcal{G}_s \cup \mathcal{G}_o, \tag{1.11}$$
$$\mathcal{G}_s = \{g_1, ..., g_{M_s}\}, \tag{1.12}$$
$$\text{and} \quad \mathcal{G}_o = \{g_{M_s+1}, ..., g_{M_s+M_o}\}; \tag{1.13}$$

4. *f_t the function which assigns each customer $u \in \mathcal{U}$, to his existing cover $f_t(u)$ at time t:*

$$f_t : \mathcal{U} \longrightarrow \mathcal{P}(\mathcal{G}), \tag{1.14}$$

where $\mathcal{G}$ is defined by Equation (1.10) and $\mathcal{P}(\mathcal{G})$ is the set containing all subsets of $\mathcal{G}$.
Since each customer must select at least one guarantee from $\mathcal{G}_s$, $\forall u \in \mathcal{U}, f_t(u) \cap \mathcal{G}_s \neq \emptyset$. We also consider $f_t(u)^\mathsf{c}$, the subset of guarantees u did not subscribe to, i.e.: $f_t(u) \cup f_t(u)^\mathsf{c} = \mathcal{G}$.

5. *Φ the function which assigns each customer u_i to the index $\Phi(u)$ of the guarantee $g_{\Phi(u)}$ suggested by the recommendation system:*

$$\Phi : \mathcal{U} \longrightarrow [\![1; M]\!]. \tag{1.15}$$

Then $g_{\Phi(u)} \in f_t(u)^\mathsf{c}$, i.e. the guarantee recommended does not belong to the current customer's cover.

To sum up, the test version of the recommendation system suggests to each customer u, who has already subscribed to the set of guarantees $f_t(u)$ at time t, to add the guarantee $g_{\Phi(u)} \in f_t(u)^\mathsf{c}$ to his insurance cover.

1.7.3 Architecture of the recommendation system tested

Therefore, each customer u from the portfolios involved in the pilot phase is attributed both a probability to accept to augment insurance cover $y_u(t)$ and the most appropriated additional guarantee $g_{\Phi(u)}$. We assume that these two outputs are independent.

Assumption 2 *For each customer u, we consider that $y_u(t)$ and $g_{\Phi(u)}$, defined by Equations (1.1) and (1.15) respectively, are independent.*

This strong assumption is motivated by the fact that the objective is to target the right amount of customers most likely to add a guarantee and to avoid non converted opportunities, instead of simply offering the next best offer for everyone. Since we focus on a small amount of covers, setting an independent algorithm to evaluate which cover/guarantee should be added to current covering allows to be accurate on predictions: the back-testing of the algorithms tested will validate this assumption *a posteriori* (see 1.7.4.1).

Figure 1.3 presents the approach to build the targeted recommendation system, which is divided in several steps, from step A to step E.

After aggregating the different data sources (step A), we perform feature engineering (step B). These steps are detailed in Section 1.3. Assumption 2 is then made, illustrated by the separation of steps C1 and C2.

Step C1 corresponds to the algorithm described in 1.5. Step C1 answers the question: to whom should we address the recommendations in priority?

Step C2 aims to predict which insurance cover/guarantee is most likely to be added, among the missing covers of the customers. This step answers the question: which additional insurance cover should we recommend? After testing several approaches, this step is performed by the Apriori algorithm.

The Apriori algorithm was introduced by Agrawal and Srikant in [61], in order to find association rules in a dataset (e.g. a collection of items bought together). The main applications of the Apriori algorithm are:

- Market basket analysis, i.e. finding which items are likely to be bought together, as developed in [61]. This technique is used by Amazon for their recommendation systems;
- Several types of medical data analysis, e.g. finding which combination of treatments and patient characteristics cause side effects. In [62], the authors show that Apriori is the best method to optimize search queries on biomedical databases, among other algorithms such as K-means or SVM (accuracy of 92% for Apriori versus 80% and 90% for K-means and SVM respectively);
- Auto-complete applications, i.e. finding the best words associated to a first sequence of words. This method is used by Google to auto-complete queries in the search engine.

Applied to our recommendation system, the Apriori algorithm could detect, from an initial set of guarantees subscribed by a customer, the guarantee most frequently associated with

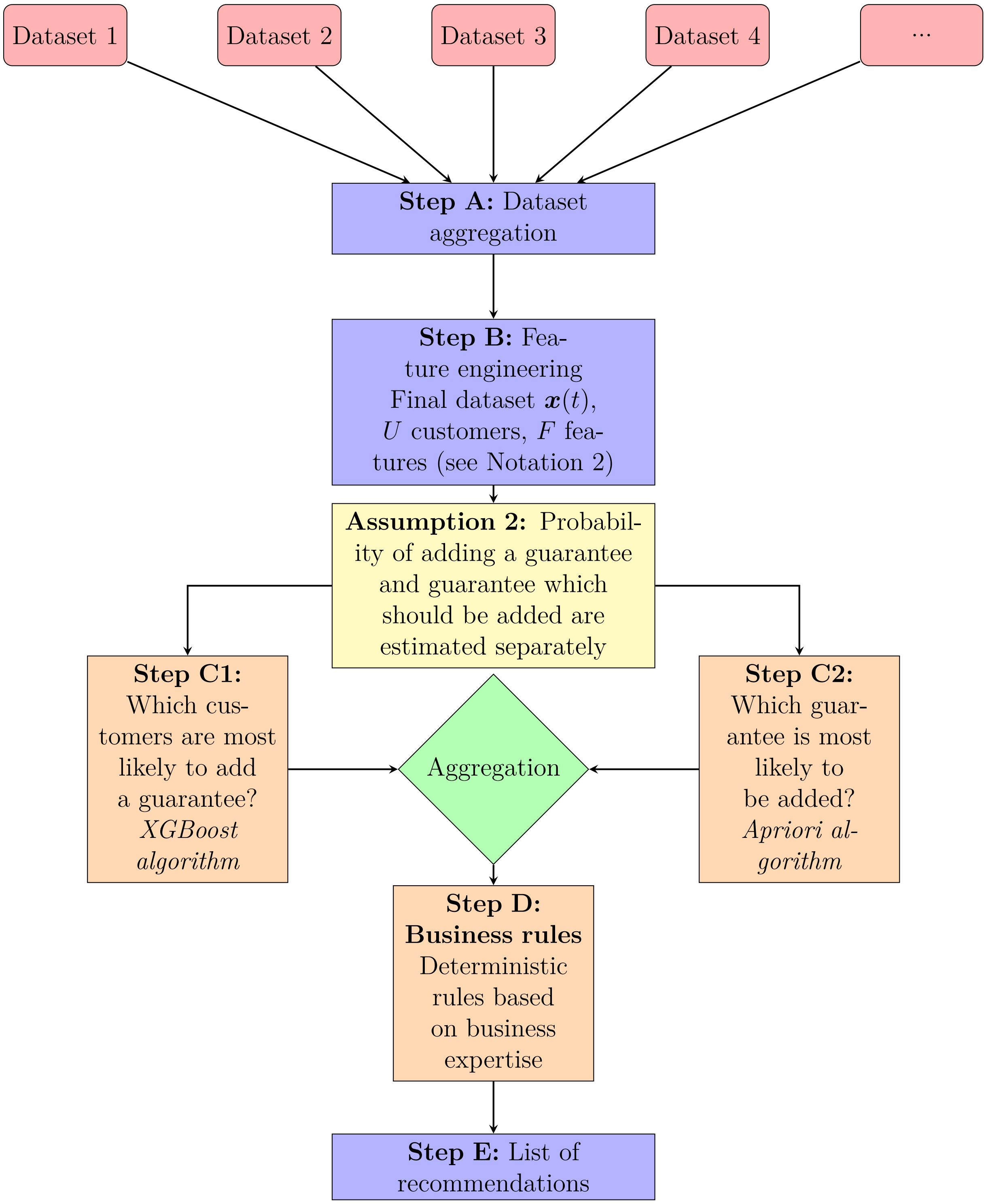


Figure 1.3: Global architecture of the recommendation system

this initial cover. Our statement is that this guarantee is most likely to be added by a customer and should be consequently the one to be recommended.

Notation 5 *An association rule R is of the form:*

$$R : R_1 = \{g_{r_1(1)}, ..., g_{r_1(N_R)}\} \rightarrow R_2 = \{g_{r_2}\}, \tag{1.16}$$

where R_1 is a set of N_R guarantees, $r_1(k)$ the index of the k^{th} guarantee of R_1 ($k \in [\![1; N_R]\!]$), R_2 a singleton of one guarantee of index r_2 which does not belong to R_1: $R_1 \bigcap R_2 = \emptyset$.

The Apriori algorithm generates every association rule $\mathcal{R}$ appearing from existing customers' covers.

Definition 4 *The set of all association rules $\mathcal{R}$ from the customers' set $\mathcal{U}$ is:*

$$\mathcal{R}_{\mathcal{U}} = \left\{R : R_1 \rightarrow R_2 \middle| \exists u \in \mathcal{U}, R_1 \cup R_2 \subseteq f_t(u)\right\}, \tag{1.17}$$

where R and $f_t(u)$ are respectively defined by equations (1.16) and (1.14).

Example 2 *If a customer subscribed to guarantees $\{g_1, g_2, g_3\}$, then association rules implied by this cover are $\{g_1, g_2\} \rightarrow \{g_3\}$, $\{g_1, g_3\} \rightarrow \{g_2\}$, $\{g_2, g_3\} \rightarrow \{g_1\}$, $\{g_1\} \rightarrow \{g_2\}$, $\{g_2\} \rightarrow \{g_1\}$, $\{g_1\} \rightarrow \{g_3\}$, $\{g_3\} \rightarrow \{g_1\}$, $\{g_2\} \rightarrow \{g_3\}$ and $\{g_3\} \rightarrow \{g_2\}$.*

For each association rule, the support and the confidence are calculated and defined below.

Definition 5 *The support S_R of a rule $R : R_1 \rightarrow R_2$ is the number of customers who subscribed to R_1 and R_2:*

$$S_R = \#\left\{u \in \mathcal{U} \middle| f_t(u) \supseteq R_1 \cup R_2\right\}, \tag{1.18}$$

where # denotes the cardinality of a set. The confidence C_R of a rule $R : R_1 \rightarrow R_2$ is the proportion of customers who subscribed to R_1 and also subscribed to R_2:

$$C_R = \frac{\#\left\{u_i \in \mathcal{U} \middle| f_t(u) \supseteq R_1 \cup R_2\right\}}{\#\left\{u \in \mathcal{U} \middle| f_t(u) \supseteq R_1\right\}}, \tag{1.19}$$

where R and $f_t(u)$ are respectively defined by Equations (1.16) and (1.14).

To define which guarantee is most likely to be added by each customer, we generate every association rule based on the Apriori algorithm. Then for each customer u, we keep the eligible rules defined below.

Definition 6 *Eligible rules ER(u) for a customer u are every association rule $R : R_1 \to R_2$ where R_1 is a subset of customer's cover and R_2 is not subscribed by u:*

$$ER(u) = \{R : R_1 \to R_2 \in \mathcal{R}_{\mathcal{U}} \Big| R_1 \subseteq f_t(u) \text{ and } R_2 \cap f_t(u) = \emptyset\}, u \in \mathcal{U}, \tag{1.20}$$

where $R_{\mathcal{U}}$ and $f_t(u)$ are respectively defined by equations (1.17) and (1.14).

Once eligible rules are filtered, we keep the association rule $R : R_1 \to R_2$ with the highest confidence, defined by (1.19). Therefore $R_2 = \{g_{\Phi(u)}\}$ is recommended:

$$\{g_{\Phi(u)}\} = R_2, \text{ such that } R : R_1 \to R_2 = \underset{R \in \mathrm{ER}(u)}{\mathrm{argmax}} C_R, \tag{1.21}$$

where $\Phi(u)$ and $\mathrm{ER}(u)$ are respectively defined by equations (1.15) and (1.20).

The entire process is synthesized in the pseudo-code in Algorithm 3.

Algorithm 3: Step C2

Inputs: Insurance cover from every customer u at time t: $f_t(u), u \in \mathcal{U}$
Result: The most appropriated additional cover for every customer u: $g_{\Phi(u)}, u \in \mathcal{U}$

```
Generate every association rule R_U thanks to the Apriori algorithm from a
  training set;
for u ∈ U do
    Compute eligible rules ER(u) for customer u;
    Compute R = argmax_{S∈ER(u)} C_S, where R : R1 → R2 is the association rule with the
      highest confidence;
    Recommend guarantee {g_Φ(u)} = R2.
end
```

Business rules (step D) consist in adding additional deterministic rules based on agent expertise and product knowledge. This step avoids computing recommendations which are not usable in practice. For instance, some customers are already covered by recommended guarantee because they subscribed to an old version of car insurance product, whose guarantees were defined differently. Some simple business rules downstream of the model take into account these particularities.

The final step E generates the list of recommendations. Once we have every couple $\{g_{\Phi(u)}, y_u(t)\}$, agents suggest guarantees for customers most likely to accept recommendations. Thus we sort this list of couples by decreasing $y_u(t)$ and we group them by agent (i.e. each customer has an assigned agent, so we generate a list of recommendations for each agent), to obtain the final list of recommendations.

1.7.4 Results

1.7.4.1 Back-testing of step C2

To evaluate the accuracy of step C2, we consider all the customers who added a guarantee in the past, and we observe the proportion of those that added the same guarantee that would have been recommended by the algorithms. We compare results of selected method, the Apriori algorithm, with other approaches for recommendation systems:

- Random: as a benchmark, we evaluate the accuracy of choosing randomly the guarantee to recommend to u from $f_t(u)^{\mathsf{c}}$, where $f_t(u)^{\mathsf{c}}$ is defined by equation (1.14).
- Popular: we recommend to u_i the most popular guarantee from $f_t(u^{\mathsf{c}}_)$, i.e. the most subscribed guarantee from $f_t(u)^{\mathsf{c}}$ in $\mathcal{U}$, where $f_t(u)$ is defined by equation (1.14).
- IBCF (see [9]): this approach estimates distances between items (i.e. the guarantees) and recommends the nearest guarantee from existing customer's cover.
- UBCF (see [8]): this approach is dual with IBCF. It estimates distances between users and recommends to a customer the guarantee that nearest users, according to this distance, subscribed to.
- SVD (see [10]): IBCF and UBCF use distances calculated from binary user-item matrix, defined by:

$$\tilde{R} = (\tilde{R}_{u,j})_{u\in\mathcal{U},j\in[\![1;M]\!]} = (\mathbb{1}_{g_j\in f_t(u)}), \tag{1.22}$$

  where $f_t(u)$ is defined by equation (1.14). SVD is a matrix factorization method which consists in decomposing R matrix described above into rectangular matrices with lower dimensions.

The results are synthesized in Table 1. They show that the Apriori approach presents the best performance on our data.

Table 1.4: Step C2 - comparison of methods

Method	Accuracy
Random	49 %
Popular	67 %
IBCF	71 %
UBCF	82 %
SVD	72 %
Apriori	95 %

To validate Assumption 2 *a posteriori*, we compared the accuracy of the Apriori algorithm on two sub-populations: customers with the lowest and those with the highest probabilities of adding a guarantee, calculated in step C1 (i.e. customers with a probability lower (respectively higher) than median probability on the whole population). The results are synthesized in Table 1.5.

Table 1.5: Validation of Assumption 2

Population	Accuracy
Low probabilities	93.4 %
High probabilities	96.6 %
Top 15% probabilities	97.1 %

The results show that for both sub-populations, the Apriori algorithm has similar accuracies, which validates (or at least does not contradict) our assumption a posteriori. The top 15% probabilities, who are the most likely to be targeted by our recommendation system and related to the 15% of customers who added a guarantee to their car insurance in the past, reach a 97.1% accuracy.

1.7.4.2 Pilot phase

Customers' profile. We present an overview of the type of customers selected by the recommendation system, compared with the global portfolio of the agents who took part in the pilot phase. Table 1.6 presents some features of these customers. This table allows us to make an archetype of customers who are more likely to add a guarantee, according to the XGBoost algorithm. For instance, the first row means that the selected customers were 2.2% younger than the average customer of the four agencies' portfolio.

Table 1.6: Profile of customers selected for the pilot phase

Characteristic	Delta (%)
Age	−2.2%
Living in Luxembourg City	+8.1%
Number of guarantees	+4.7%
Car insurance premium	+15.1%
Number of products	+27.4%
Insurance premium	+12.9%
Number of vehicles	+10.1%
Age of vehicles	−6.4%
Price of vehicles	+33.5%
Scoring	+0.5 level (on a scale from 1 to 5)
Number of amendments	+11.1%

We could particularly note that selected customers have less guarantees than average, more products and more vehicles subscribed at Foyer, more expensive and more recent cars. These observations make sense: the XGBoost algorithm targets customers who have a reduced cover compared to their current car and their purchasing power. Besides, we could notice that selected customers have a better scoring and a higher number of amendments on their contracts: it shows that our recommendation system targets customers of better quality and those who decided to modify their contract in the past.

Global results. Overall acceptance rate is 38%. It is below expectations from back-testing, as shown in Table 1.7 (see below). This could be explained by the fact that back-testing is made on past guarantees additions, instead of past recommendations.

However this result remains promising since benchmark acceptance rate for such marketing campaigns is about 15% (see Remark 1 below for more details). This standard rate comes from previous results of a similar test based on the recommendation system developed in [63].

Remark 1 *It is worth mentioning that the acceptance rates from our study and from [63] are both from a selection of the global portfolio of customers. Our acceptance rate is calculated on the top 10% recommendations according to the XGBoost algorithm, among the portfolio of the four agents. In [63], 366.998 recommendations are calculated; 737 received agent action (0.02%) and 104 were accepted (14% of the recommendations managed by agents).*

Another benchmark could be classic up-selling campaigns from Foyer, which have a conversion rate from 5% to 10%. [63] also mentions that the standard industry conversion baseline is 12%. Thus, specific targeting allowed agents to increase significantly accuracy of their up-selling actions, even if acceptance rate is lower than back-testing results.

Results by agent. Table 1.7 shows expected acceptance rate from back-testing and actual acceptance rate by agent, observed during one year.

Table 1.7: Pilot phase - acceptance rate by agent

Agent	Expected acceptance rate	Actual acceptance rate
Agent 1	61 %	39 %
Agent 2	58 %	48 %
Agent 3	60 %	39 %
Agent 4	52 %	21 %
Overall	57 %	**38** %

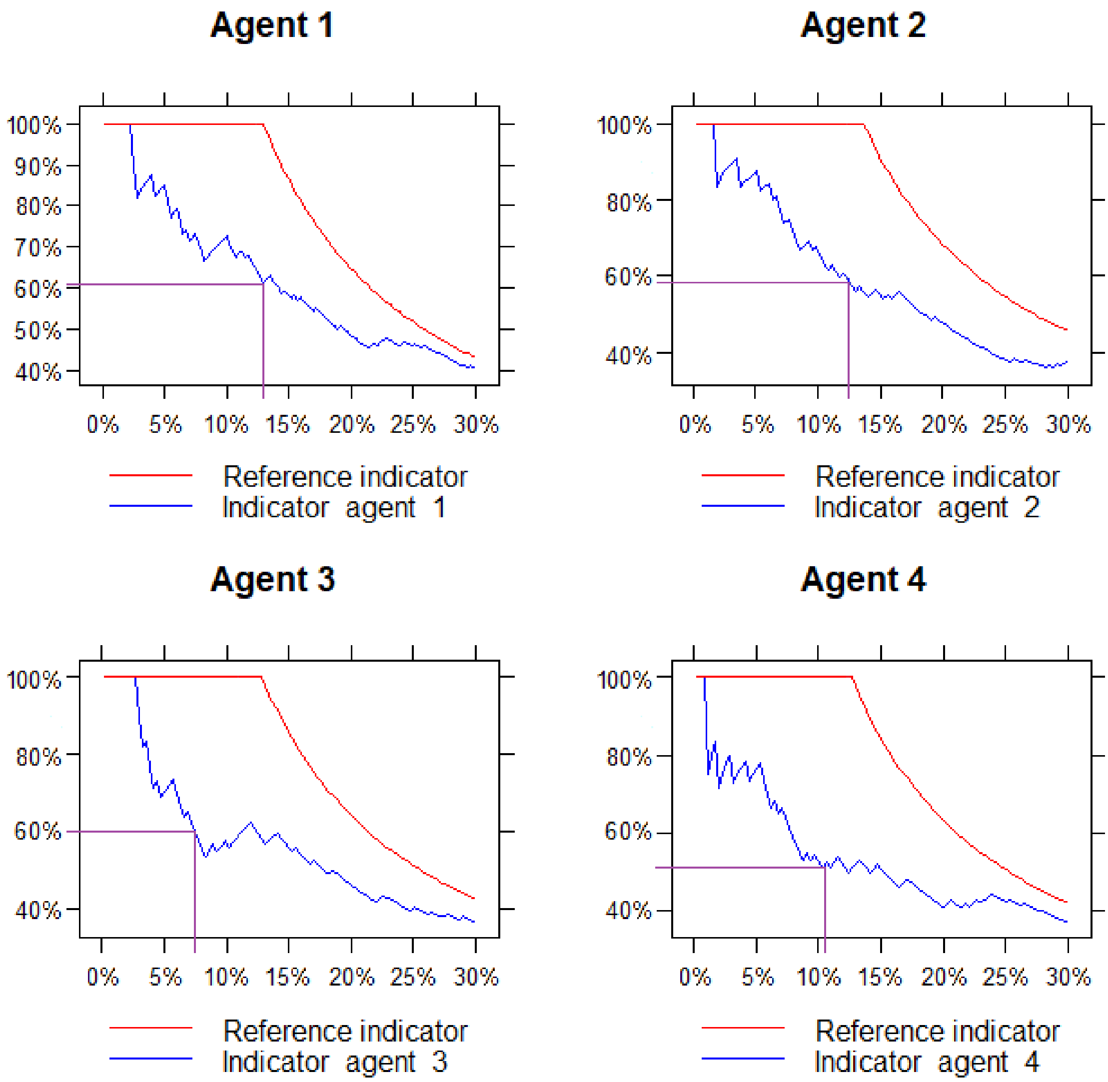


Figure 1.4: Back-testing of step C1, on agents participating to pilot phase

As mentioned for global results, acceptance rates are below expectations from back-testing results for every agent. However, these back-testing results allowed us to detect that Agent 4 should have a lower acceptance rate before the pilot phase.

Results by guarantee. Table 1.8 presents the distribution of guarantees recommended and the acceptance rates associated.

Table 1.8: Pilot phase - distribution and acceptance rate of guarantees recommended

Guarantee	Percentage of recommendations suggesting the guarantee	Acceptance rate
Driver injury	58 %	47 %
Glass damage	24 %	23 %
Legal cover	12 %	33 %
Road accident	6 %	26 %
Overall	100 %	**38** %

This table reveals that driver injury guarantee is by far the most recommended and the most accepted by customers as well. However, the less commonly recommended guarantees have a promising acceptance rate (not below 23%, which is higher that our 15% benchmark), which shows that the recommendation system could be efficient for different types of needs from customers.

1.7.4.3 Agents' feedback

The main reasons why some customers did not accept their recommendations, according to agents' feedback, are the following:

- Customers only subscribed to essential guarantees and do not want to spend more money on their car insurance,
- Customers already subscribed to the same type of recommended guarantee in another company. For instance, some customers already have legal cover from their employer.

Some recommendation refusals could also have been avoided because agents already suggested the guarantee to customers in the near past, without any mention about this exchange in Foyer's datasets. This inconvenience will be rectified subsequently. Besides, agents suggested that associating an explanation to every commercial opportunity would improve the recommendation system.

Remark 2 *During the pilot phase, we observed that the acceptance rate decayed through time. It is explained by the fact that agents dealt first with recommendations for which they were pretty sure that they will get a positive answer, due to their knowledge of their customers. Given this observation, we could think that the recommendation system is useless since agents already knew the main part of successful recommendations. But agents highlighted the fact that one of the advantages of the recommendation system is to analyse all their portfolio equally, which allowed them to remind some customers that they would not have dealt with at the precise moment of the campaign. They also reported that some recommendation were surprisingly accepted against their intuition.*

Chapter 2

Influence of life events: Hawkes processes and properties

2.1 Objective

In Chapter 1, we built a recommendation system depending on data excluding life events was built. As introduced in Assumption 1 and Equation (2), this chapter focuses on the model which takes into account customer's life events history, in order to update the recommendations whenever an event occurs.

The necessity to include this temporal dimension leads naturally to point processes, which are a category of temporal models specifically suited to the occurrences of events in time and space. Moreover, we have to model events that are temporally clustered, i.e. the occurrence of a life event makes the event "Subscription to a new insurance cover" more or less likely to occur in the near future. That is why we focus on MHP Multivariate Hawkes Processes. In this chapter, a MHP framework is developed so that the model is suited to data about life events and to the objective of providing personalized recommendations for customers. The framework includes the MHP definition, mathematical properties, and estimation and simulation methods for MHP specifically designed for our purpose.

Chapter 2 is organized as follows. Section 2.2 introduces the notion of point processes and defines both univariate and multivariate Hawkes Processes by fixing all the notations needed briefly introduced in Section 0.4. Section 2.3 presents the main assumptions about MHP building. Section 2.4 provides an algorithm to estimate MHP parameters in function of data particularities. Section 2.5 states several mathematical properties of the MHP specifically designed for the recommendation system. Proofs are in Section 2.6.

2.2 Hawkes processes definition

2.2.1 Point processes

In the context of the recommendation system, a point process is a set of points representing events randomly located on the time line. Concerning global theoretical notions about point processes, we refer to [64] which is a reference in this area.

Definition 7 ***(Point process).*** *A point process* $\mathbf{T}$ *is an increasing sequence of random variables* $\mathbf{T} = \{T_1, T_2, ...\}$ *called occurrences which takes values in* $[0, +\infty[$*, such that* $\mathbb{P}(0 < T_1 < T_2 < ...) = 1$*.*

We introduce a formal definition of the history of occurrences, which contains all the information about observed events.

Definition 8 ***(History of occurrences).*** *We define* $(\mathcal{H}(t), t \geq 0)$ *the filtration of the history of the occurrences until time* t*.*

A mathematical interpretation of a point process is the notion of counting process.

Definition 9 ***(Counting process).*** *We say that* N *is a counting process if it is an almost surely finite stochastic process and a right-continuous step function with +1 increments after each step, taking values in* $\mathbb{N}$*, and* $N(0) = 0$*.*

The two notions of point process and counting process are interchangeable. Indeed, a counting process could be seen as the cumulative count of a point process. The link between $\mathbf{T}$ and N is:

$$\forall t \in \mathbb{R}_+, N(t) = \#\Big\{T_i \in \mathbf{T} \Big| T_i \leq t\Big\}, i \geq 1,$$

where # denotes the cardinality of a set and T_i is the time of the i^{th} occurrence in the time interval $[0, T]$, where T is the final time of observation.

2.2.2 Characterization by conditional intensity function

The most frequently used characterization of a Hawkes process is the conditional intensity function, $\lambda(t|\mathcal{H}(t))$, introduced in Definition 1.

The following proposition is inspired from Proposition 2.2 of [65] and states that the conditional intensity function uniquely defines the point process. Thus in the following we use this characterization.

Proposition 1 *Let us consider a conditional intensity function $\lambda^*(\cdot)$, defined for any time period $[u,t]$, $0 \leq u \leq t$, such that:*

- $\lambda^*(\cdot)$ *non-negative and integrable on* $[u,t]$,
- $\lambda^*(\cdot)$ $\mathcal{H}(t)$*-measurable,*
- $\lim_{t\to+\infty} \int_u^t \lambda^*(s)ds = +\infty$.

Then there exists an unique point process with $\lambda^*(t)$ conditional intensity function.

2.2.3 Goodness-of-fit

In this subsection, we describe a procedure to evaluate the goodness-of-fit of a Hawkes process, i.e. how well it fits to a set of data. The following proposition from [64], called residual analysis, allows us to test the quality of the model.

Proposition 2 ***(Residual analysis).*** *We consider a sequence of occurrence times $\{t_1, t_2, ...\}$ and a monotonic, continuous compensator $\Lambda(t) = \int_0^t \lambda^*(s)ds$ such that:*

$$\lim_{t\to+\infty} \Lambda(t) = +\infty, \tag{2.1}$$

almost surely. The sequence $\{\Lambda(t_1), \Lambda(t_2), ...\}$ is a Poisson process with an unit rate if and only if $\{t_1, t_2, ...\}$ is a realisation from the point process defined by $\Lambda(\cdot)$.

Thanks to this proposition, we could test with many procedures whether our data fit with a Hawkes process. The previous proposition shows us that testing whether $\{t_1, t_2, ...\}$ follows a Hawkes process with intensity function λ is equivalent to test whether $\{\Lambda(t_1), \Lambda(t_2), ...\}$ form a Process process of parameter 1. It is also equivalent to test whether every interarrival time $\{\Lambda(t_1), \Lambda(t_2) - \Lambda(t_1), \Lambda(t_3) - \Lambda(t_2), ...\}$ are independent and identically distributed and follow an Exponential law of parameter 1. This could be done by:

- A Quantile-Quantile plot (QQ-plot), which represents quantiles of both distributions;
- Performing a Kolmogorov-Smirnov test, whose statistic D is the maximum absolute difference between the two distributions.

2.2.4 Univariate Hawkes processes

2.2.4.1 Definition

There exists several well-known types of point processes (see [64]).

The homogeneous Poisson point process is the simplest of them: it is characterized by a constant conditional intensity function. It has two main properties: if we consider two disjoint intervals, the numbers of events in these intervals follows a Poisson distribution and are independent.

Inhomogeneous Poisson process have a deterministic conditional intensity function which depends on time. In 1971, Alan G. Hawkes introduced in [66] the category of point processes named after him. As opposed to Poisson processes, Hawkes processes are characterized by a random intensity function, whose value depends on the history of occurrences (see Definition 8) indeed.

We propose a definition of univariate Hawkes processes, inspired from [67].

Definition 10 ***(Hawkes process).*** *Let us consider $\lambda > 0$, the background intensity, and $\mu : [0, +\infty[\to [0, +\infty[$, the triggering function. We denote $\{t_1, .., t_{N(t)}\}$ the sequence of past occurrences until time t. A point process is a Hawkes process if its conditional intensity function is of the form:*

$$\lambda^*(t) = \lambda + \int_0^t \mu(t-u)dN(u) = \lambda + \sum_{k=1}^{N(t)} \mu(t - t_k), \tag{2.2}$$

where $N(\cdot)$ is defined in Definition 9.

Hawkes processes are point processes who have a so-called "self-exciting" property. It means that each occurrence increases the intensity, according to the triggering function. A higher intensity leads to more occurrences, leading to a higher frequency, etc.

A Hawkes process with a null triggering function is a homogeneous Poisson process of rate λ. Hawkes processes are frequently studied in recent scientific literature and are used for several applications: earthquake modelling [68], criminology [69], finance [70], etc.

Since recently, insurance companies are developing an interest for Hawkes processes. They are used for calculating Solvency Capital Requirements and modelling different indicators of risks, such as ruin (improvement of Cramer-Lundberg model, see [71]) or cyber-attacks (see [72]). Hawkes processes model claims arrival, considered to follow a Poisson process in classic approaches.

Among these references, the most frequent triggering functions are exponential (used as the main example of triggering function in the original papers of Hawkes because of simplicity of calculation) and power-law functions. We propose to define Γ-Hawkes processes, based on a Gamma density as triggering function.

Definition 11 ***(Γ-Hawkes processes).*** *We define the Γ-Hawkes processes, a Hawkes process whose triggering function is of form:*

$$\mu(t) = \alpha \frac{t^{k_1-1}\exp(-\frac{t}{k_2})}{k_2^{k_1}\Gamma(k_1)}, t \geq 0, \alpha \in \mathbb{R}_+^*, k_1, k_2 \in \mathbb{R}_+^*, \tag{2.3}$$

where $\Gamma(\cdot)$ *is the Gamma function.* k_1 *is the scale parameter and* k_2 *is the shape parameter.*

Remark 3 *For* $k_1 = 1$*, we obtain the exponential triggering function:*

$$\mu(t) = \alpha \frac{\exp(-\frac{t}{k_2})}{k_2}, t \geq 0, \alpha \in \mathbb{R}_+^*, k_2 \in \mathbb{R}_+^*, \tag{2.4}$$

used for instance originally by Hawkes [66]. In this situation:

$$\lambda^*(t) = \lambda + \sum_{k=1}^{N(t)} \alpha \frac{\exp(-\frac{t-t_k}{k_2})}{k_2}. \tag{2.5}$$

For $k_1 > 1$, this triggering function allows to avoid a discontinuity in the intensity function when an event occurs, and introduce a delay between an event and its impact on the process. Indeed, in that case, $\mu(0) = 0$. This property is very helpful when considering events for which an occurrence and the resulting self-excitation are delayed. For instance, considering an epidemic, there is a delay between a people getting infected (the event "infection" occurs) and this person being infectious (the fact that the person is infectious increases the probability to have more infections). For the recommendation system purpose, we expect that there is a delay between a life event occurrence and its influence on a potential subscription to a new insurance cover.

2.2.4.2 Illustration: COVID-19 dynamics

In order to illustrate the interest of univariate Hawkes processes, we tested several modellings on real data based on this category of point process. Even if the topic is not directly linked to the recommendation system, we present in this subsection a modelling of COVID-19 dynamics by an univariate Hawkes process, because:

- This study allows us to confirm that Hawkes processes are well suited for modelling a phenomenon with self-exciting properties. Indeed, the more people are infected, the more people are likely to become infected;
- COVID-19 pandemic has a tremendous impact on insurance industry. The pandemic and the subsequent lockdown around the world lead to a global recession which affected insurance companies by increasing claims and decreasing profits (see [73]).

This subsection is inspired from [4], where we proposed to model the COVID-19 propagation dynamics with an univariate Hawkes process. The two objectives of this work were to:

- Understand COVID-19 contagion dynamics in order to make people aware of the seriousness of the situation at the beginning of the pandemic;
- Illustrate the interest of univariate Hawkes processes on real data.

Several scientific papers focused on the subject in order to propose solutions to slow down the pandemic. For instance, at the same period (beginning of the pandemic), the study in [74] uses a Poisson autoregressive model to fit the evolution of the outbreak in China, Iran, Italy, and South Korea.

We fitted the process on the number of cases observed in France over the first weeks of the pandemic (until March 8 2020), then we simulate over ten days to back-test the model (until March 16).

We have applied the model to the data from the GitHub repository of the Center of Systems Science and Engineering (CSSE) from Johns Hopkins University, in the folder dedicated to COVID-19 data [75], where we could find the time series of confirmed cases in France from January 22 to March 16.

Step 1: fitting on data until March 8. We learn our model on data observed until March 8. Parameters λ, α, k_1 and k_2 are estimated by maximizing the likelihood of the $\Gamma-$Hawkes process, which could be written only in function of $\lambda^*(.)$, thanks to an Expectation-Maximization (EM) algorithm suited for Hawkes processes implemented in [30] and described in [76].

Table 1 shows the values of estimated parameters:

Parameter	**Estimated value**
λ	0.1616
α	1.0518
k_1	1.0031
k_2	4.4141

Table 2.1: Estimated value of the $\Gamma-$Hawkes process parameters

The fact that k_1 is close to 1 means that the delay between the infection and the contagion is very short (there is no consensus on this question among scientists, the delay could vary from 0 to 10 days). Indeed it means that Gamma density is close to an exponential function. Consequently, each new case leads to a maximal jump of value α for the conditional intensity function. The fact that λ (*background intensity*) is lower than α (*exciting* part) means that the spread is mainly due to transmission from French people to French people, rather than infections from foreign countries, even if a first infection from outside is necessary to launch the outbreak.

To test whether $\{t_1, t_2, ...\}$ follows a HP with intensity function $\lambda^*(.)$, we use the QQ-plot in Figure 2.1 page 43 and we perform a Kolmogorov-Smirnov test to check whether $\{\Lambda(t_1), \Lambda(t_2), ...\}$ form a Process process of parameter 1 (see Section 2.2.3):

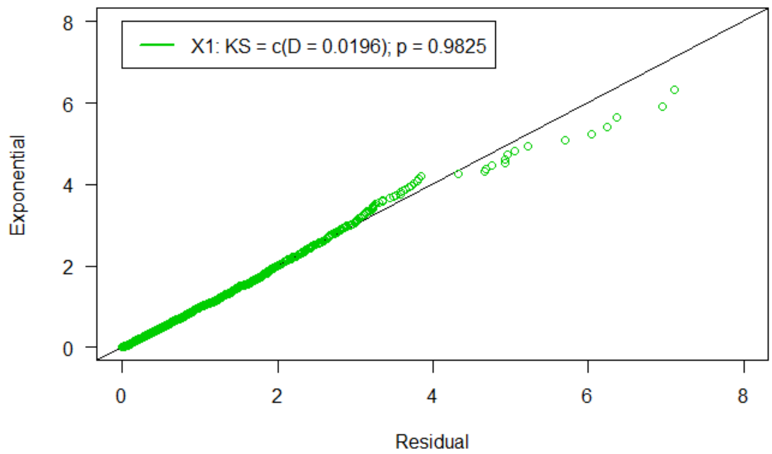


Figure 2.1: QQ-plot to check the goodness-of-fit of the model

The QQ-plot allows us to assume that the time series could be modelled by a HP. Moreover, we performed a Kolmogorov-Smirnov test on the two distributions. The maximal distance is $D = 0.0198$, with a p-value of $p = 0.9825$, which confirms our assumption.

Remark: Figure 2.2 shows the evolution of the number of infections in France from January 22 to March 8 (right Y-axis), and the conditional intensity function (left Y-axis):

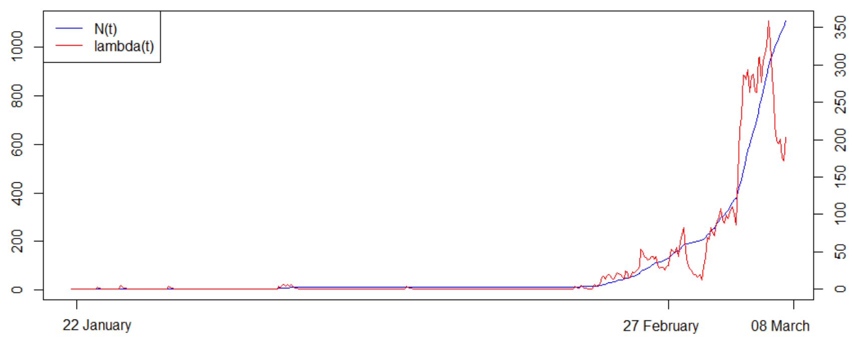


Figure 2.2: Number of infections in France from January 22^{nd} to March 8, and estimated conditional intensity function

The very first little peak of the conditional intensity function at January 24 corresponds to the first case in France. The high value of the conditional intensity function at March 8 shows that the outbreak is growing fast.

Step 2: back-testing on data from March 8 to March 16. To back-test our model, we simulate the evolution of the outbreak from March 8 to March 16, then we compare with real data. We simulate the HP thanks to Ogata's thinning algorithm [77]. The idea of Ogata's algorithm is that randomly removing points from a "faster" simulated Poisson process (i.e. with a higher intensity function than targeted Hawkes process) allows to simulate a Hawkes process.

Figure 2.3 shows the comparison between the simulation and the real data:

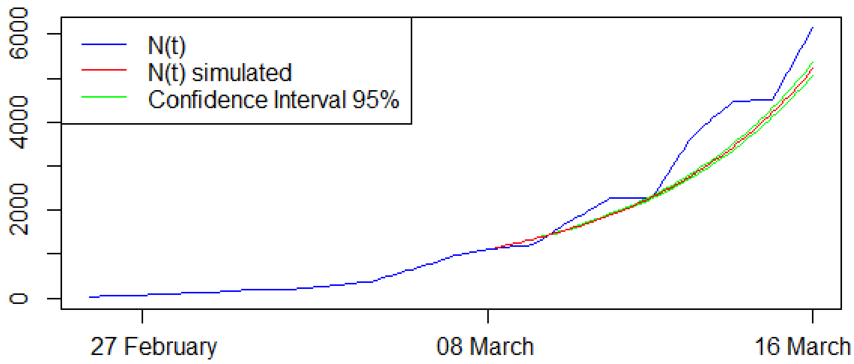


Figure 2.3: Simulation of the number of infections between March 8 and March 16

Figure 2.3 shows that the model catches the trend of the outbreak evolution, and validates our approach. At March 16, the estimation of the number of cases is 5,208.96. The confidence interval of 95% (CI 95%) is $[5,208.96 - 140; 5,208.96 + 140] = [5,068.96; 5,348.96]$. The actual number of infected people is 6,130.

Step 3: simulation until March 26 according to two scenarios. The last step simulates the number of infections ten days later (Thursday March 26). We focus on two scenarios: either the dynamics of the outbreak remains the same, or French government takes measures similar to what Chinese authorities decided to.

Scenario 1: dynamics remain the same. In a first scenario, we estimate of the number of infections, if the dynamics of the outbreak remain the same, in ten days (Thursday March 26), that's to say if French people have not changed their social interactions in the first weeks of March.

Scenario 2: France shuts down the country like China partially did. In a second scenario, we assume the French government takes similar measures than China at March 16. To take into account these decisions, we adapt our HP model into a non-linear form:

$$\lambda^*(t) = \Big[\lambda + \sum_{k=1}^{N(t)} \mu(t - t_k)\Big] \times \min\Big(1, \Big(\frac{t}{t_m}\Big)^a\Big), \tag{2.6}$$

where t_m is the time when the measures are taken (i.e. March 16) and $a > 1$ is a parameter to estimate. This model allows the conditional intensity function to be inhibited from t_m, thanks to the inhibiting part $\min\Big(1, \Big(\frac{t}{t_m}\Big)^a\Big)$.

To estimate a, we focus on data from the Chinese province of Jiangxi, which is close to the province of Hubei, the beginning of the outbreak. We fit a non-linear Hawkes process for these data. We denote:

$$\lambda_c^*(t) = \Big[\lambda_c + \sum_{k=1}^{N_c(t)} \mu_c(t_c - t_{c,k})\Big] \times \min\Big(1, \Big(\frac{t}{t_{c,m}}\Big)^{a_c}\Big), \tag{2.7}$$

where index c refers to China, and then we will assume that $a_c = a$. To fit this model, we first fix $t_{c,m} =$ February 5, which is the approximate date from when China decided to shut down the country. Then, we fit the parameters with data until February 5, so that the inhibiting part is equal to 1. After getting the parameters λ_c and μ_c, we simulate the HP from February 5 to February 20 with a range of values for a_c. We choose the value of a_c which gives the estimated number of cases which is the closest to the actual number of cases at March 16. Figure 2.4 represents the simulation of the fitted non-linear HP and the actual number of cases in Jiangxi, with the final choice for a_c parameter: $a_c = -1.403$.

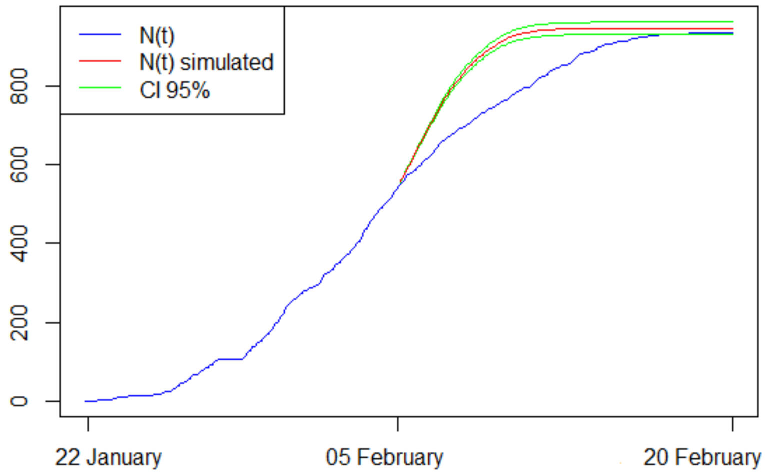


Figure 2.4: Calibration of a by simulating the number of infections between February 5 and February 20 for Chinese data

Then for the French process, we do the same simulation as the first scenario, but with the non-linear HP and with the parameters t_m = March 16 and $a = a_c = -1.403$ for the inhibiting part. Figure 2.5 shows the comparison between the first and the second scenario.

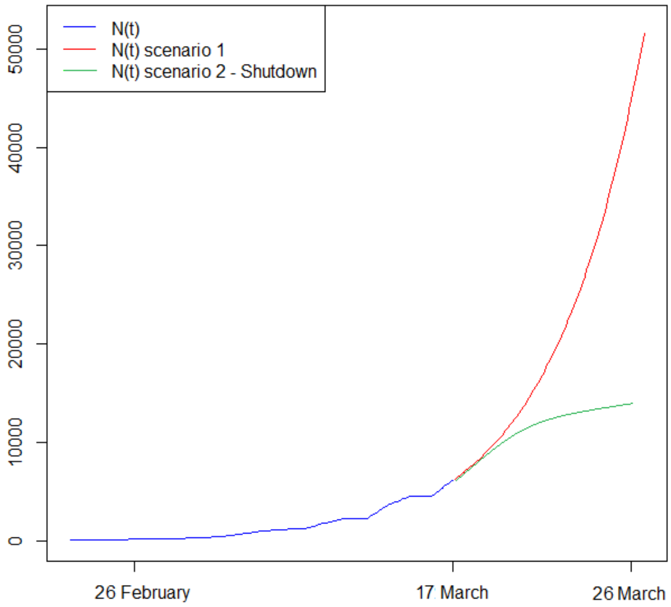


Figure 2.5: Simulation of the number of infections between March 16 and March 26 for the two scenarios

It shows that taking measures as in China would reduce significantly the number of estimated cases. We obtain an estimated number of cases of 51,642 cases in the first scenario, and 14,621 cases in the second scenario, approximately three times less!

But data we have worked on are just the number of official cases. We could reasonably think that the actual number of cases was much higher, because hospitals don't have enough doctors and material to test every people who would want to. Having COVID-19 symptoms was not enough to be tested (see [78] for instance). According to [79], 11,895 tests were performed on French population at March 9, for 1,209 official cases, that's to say a rate of 10.1% positive tests. Let's assume that this rate is the same for our simulation. It means that $\frac{51{,}642}{10.1\%} = 511,306$ tests would have been performed, that's to say only 0.7% of the population! Consequently, the actual number of cases would be much higher that 18,529. It is hard to quantify, but we could imagine that doctors could only focus on one case of 3, or one case of 5.

This study was published in March 2020, we could now compare the two scenarios with the actual number of infected people. There were 51,642 cases in the first scenario, and 14,621 cases in the second scenario. The real figure is 28,869 cases, which is approximately the average of the two scenarios. This could be explained by the fact that at this point of the pandemic people were already aware that they should limit their exposition to the virus, but they did it spontaneously and less efficiently than an actual quarantine.

Conclusion. As a conclusion of this short study, our model with a Hawkes process shows that COVID-19 seemed to have a *self-exciting* dynamics at the beginning of the pandemic. The simulations confirm that the evolution of the outbreak would have been exponential if people did not change their habits.

2.2.5 Multivariate Hawkes processes

The global objective is to model how life events affect the propensity for a customer to modify his cover. Univariate Hawkes processes only model the influence of an event on himself, not on other events. Multivariate Hawkes Processes are point processes where an occurrence for one type of event modify the probability of the other events to happen.

2.2.5.1 Definition

In this section, we introduce the notations and definitions needed for the construction of our Multivarate Hawkes Process. First, we introduce some notations which are necessary to define the Multivariate Hawkes Process.

Notation 6 *We denote:*

1. m *the total number of life events considered (see Definition 12);*
2. T_u *the time length we observe life events for customer* u*;*
3. $\mathcal{H}_u(t)$ *the list of life events which occurred to customer* u *until time* t*;*
4. $N^u(t)$ *the counting process which corresponds to the total number of events observed for customer* u *from time* 0 *to time* t*;*
5. $N_i^u(t)$ *the counting process which corresponds to the total number of events of type* i *observed for customer* u *from time* 0 *to time* t*. We have* $N^u(t) = \sum_{i=1}^m N_i^u(t)$.

We build a Multivariate Hawkes Process which will be defined by m conditional intensity functions, one per type of event, instead of one as in the univariate case. For each event, there exists not only a "self-exciting effect" (i.e. an occurrence of event i could influence the intensity of event i), but also a "mutual exciting effect" (i.e. an occurrence of event i could influence the intensity of any event $j, j \in \{1, ..., m\}$). For each customer $u, u \in \mathcal{U}$, we consider the counting processes $\{N_i^u(\cdot), i \in \{1, ..., m\}\}$, where $N_i^u(\cdot)$ refers to the type of event number i and where each type of event i could influence any other event $j, j \in \{1, ..., m\}$, including $j = i$.

We define a Multivariate Hawkes Procces in Definition 3. Therefore, $\lambda_1^{u,*}(t)$ represents the propensity for customer u to add an insurance cover at time t; this quantity will be integrated to the recommendation system. The probability that the customer would add a guarantee/product is influenced by $\lambda_1(\boldsymbol{x_u})$, which models the impact of the characteristics of the customer (*a priori* vision), and by the sum of $\mu_{1,j}(t - t_j^k)$, which models the impact of the k^{th} event of type j, occurred at time t_j^k (*a posteriori* vision).

2.2.5.2 Simulation

In order to simulate the estimated Multivariate Hawkes Process for customer u from time 0 to T_u, we use the Ogata's thinning algorithm (see [77]), written in Algorithm 4 by taking into account all the notations introduced previously.

In order to simulate an inhomogeneous Poisson process, the method of thinning consists in simulating a "faster" homogeneous Poisson process (i.e. with an intensity which is an upper bound of the intensity of the process to simulate), then removing probabilistically several arrival points (see [67]). Ogata's algorithm is based on the same idea. Since the intensity of a Hawkes process increases at each arrival, we could find an upper bound of this intensity between two arrivals only.

2.3 Main assumptions on MHP

For the purpose of the recommendation system, we make assumptions about the background intensity $\lambda_i(\boldsymbol{x_u}), i \in \{1, ..., m\}$ and the triggering functions $\mu_{i,j}, i, j \in \{1, ..., m\}$, introduced in Definition 3.

2.3.1 Background intensity

For each event i, we aim to estimate the background intensity of customer u, $\lambda_i(\boldsymbol{x_u})$, where $\boldsymbol{x_u}$ is a set of features describing the customer u. In our approach, we model for the event i the background intensity of customer u, described by $\boldsymbol{x_u}$, as follows:

$$\lambda_i(\boldsymbol{x_u}) = \mu_i f_i(\boldsymbol{x_u}), \tag{2.8}$$

where:

- μ_i is a proportional factor, common for each customer u. We denote $\boldsymbol{\mu} = \left(\mu_i\right)_{i\in\{1,\dots,m\}}$;
- f_i is a function which, to each data $\boldsymbol{x_u}$, associates $f_i(\boldsymbol{x_u}) \in \mathbb{R}$ which reflects the propensity for an event i to occur for a customer u, excluding the triggering of other events.

Algorithm 4: Ogata's thinning algorithm

Result: Simulated sequence of events for customer u: $\{(e_k^{u,sim}, t_k^{u,sim})_{k=1}^{N^{u,sim}}\}$

1. Generate the first event:

$k = 1$;

$\lambda_{max} = \sum_{i=1}^{m} \lambda_i^{u,*}(0)$;

Simulate $V \sim \mathcal{U}[0,1]$;

$s = -\frac{1}{\lambda_{max}} \ln(V)$;

if $s > T_u$ **then**

  Go to step 3.;

else

  Simulate $W \sim \mathcal{U}[0,1]$;

  $e_k^{u,sim} = j$ such that $\sum_{i=1}^{j} \lambda_i^{u,*}(s) \leq \lambda_{max} \times W \leq \sum_{i=1}^{j+1} \lambda_i^{u,*}(s)$;

  $t_k^{u,sim} = s$

end

2. General:

$k = k + 1$;

$\lambda_{max} = \sum_{i=1}^{m} \lambda_i^{u,*}(t_{k-1}^{u,sim}) + \sum_{i=1}^{m} a^{sum}_{e_{k-1}^{u,sim},i}$ (see Section 3.3.1)

Simulate $V \sim \mathcal{U}[0,1]$;

$s = s - \frac{1}{\lambda_{max}} \ln(V)$;

if $s > T_u$ **then**

  Go to step 3.;

else

  Simulate $W \sim \mathcal{U}[0,1]$;

  if $\lambda_{max} \times W \leq \sum_{i=1}^{m} \lambda_i^{u,*}(s)$ **then**

    $e_k^{u,sim} = j$ such that $\sum_{i=1}^{j} \lambda_i^{u,*}(s) \leq \lambda_{max} \times W \leq \sum_{i=1}^{j+1} \lambda_i^{u,*}(s)$;

    $t_k^{u,sim} = s$

  else

    $\lambda_{max} = \sum_{i=1}^{m} \lambda_i^{u,*}(s)$ and simulate another time s;

  end

end

3. Output: $\{(e_k^{u,sim}, t_k^{u,sim})_{k=1}^{N^{u,sim}}\}$

Through this modelling, each customer's background intensity depends on his characteristics.

2.3.2 Triggering functions

We assume that the triggering functions are a linear combination of basis functions, in order to take into account several delays of impact between an event and its consequences on other events.

Notation 7 *We denote:*

- D *the number of basis functions;*
- g_d *the* d^{th} *basis function,* $d \in \{1, ..., D\}$*;*
- $\boldsymbol{A_d} = \left(a_{i,j}^d\right)_{i,j=1,...,m}$ *the* d^{th} *infectivity matrix, where* $a_{i,j}^d$ *captures the impact of event* j *on event* i *relatively to the* d^{th} *basis function;*
- $\boldsymbol{A} = \{\boldsymbol{A_d}\}_{d\in\{1,...,D\}}$ *the set of the* D *infectivity matrix.*

Therefore, the triggering function $\mu_{i,j}$ that models the effect of event j on event i follows the equation below:

$$\mu_{i,j}(t) = \sum_{d=1}^{D} a_{i,j}^d g_d(t). \tag{2.9}$$

We propose the following basis functions for our data:

$$g_d(t) = \begin{cases} \frac{1}{\sqrt{2\pi}\omega} \exp(-\frac{t^2}{2\omega^2}) & \text{(Gaussian density) if } d = 1; \\ \frac{t^{\kappa-1}}{(d\omega)^\kappa \Gamma(\kappa)} \exp(-\frac{t}{d\omega}) & \text{(Gamma density) otherwise,} \end{cases} \tag{2.10}$$

where ω and κ are respectively scale and shape parameters for the Gamma density. The first basis function is inspired from [80] to take into account immediate triggering (since the triggering function is strictly positive at time $t = 0$), while we sketched the main interests of Gamma density as triggering function for insurance data in [2] (since the triggering effect is delayed in time because the function is null at time $t = 0$).

2.4 Estimation

2.4.1 Objective function

We denote Θ the parameters of the model to estimate: $\Theta = (\boldsymbol{\mu}, \boldsymbol{A})$.

The objective function of the problem is the sum of the Negative Log-Likelihood and regularizers described below:

- **Negative Log-Likelihood:** denoted NLL(Θ), it equals:

$$\begin{aligned}\mathrm{NLL}(\Theta) = & \sum_{u=1}^{U}\Big\{\sum_{i=1}^{m}\Big(T_u\lambda_i(\boldsymbol{x_u}) + \sum_{k=1}^{N^u}\sum_{d=1}^{D} a^d_{i,e^u_k} G_d(T_u - t^u_k)\Big) - \sum_{k=1}^{N^u}\log\Big(\lambda_{e^u_k}(\boldsymbol{x_u}) \\ & + \sum_{j=1}^{k-1}\sum_{d=1}^{D} a^d_{e^u_k,e^u_j} g_d(t^u_k - t^u_j)\Big)\Big\},\end{aligned} \tag{2.11}$$

  where $G_d(T_u - t^u_k) = \int_0^{T_u - t^u_k} g_d(t)dt$;

- **Sparsity:** we expect that for couples of events which have no influence on each other, the infectivity matrix has zeros for the corresponding coefficients. That is why we add a sparsity constraint, denoted R_1:

$$R_1(\Theta) = \alpha_1\Big(\sum_{d=1}^{D}\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(a^d_{i,j})^2}{2}\Big), \tag{2.12}$$

  where α_1 is a parameter to control the influence of this regularization;

- **Minimizing the influence of overlapping events:** for some events, the date of occurrence that is in the data-set may be delayed of several days, because of administrative issues. Therefore, we could believe that an event 1 is triggered by an event 2, while event 2 actually occurred after event 1. Therefore we propose to penalize the squared sum of coefficients $a^d_{i,j}$ and $a^d_{j,i}$, in order to correct the influence of overlapping events:

$$R_2(\Theta) = \alpha_2\Big(\sum_{d=1}^{D}\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(a^d_{i,j} + a^d_{j,i})^2}{2}\Big), \tag{2.13}$$

  where α_2 is a parameter to control the influence of this regularization.

Therefore, the objective function denoted F is:

$$\begin{aligned}F(\Theta) = & \ \mathrm{NLL}(\Theta) + R_1(\Theta) + R_2(\Theta) \\ = & \sum_{u=1}^{U}\Big\{\sum_{i=1}^{m}\Big(T_u\lambda_i(\boldsymbol{x_u}) + \sum_{k=1}^{N^u}\sum_{d=1}^{D} a^d_{i,e^u_k} G_d(T_u - t^u_k)\Big) - \sum_{k=1}^{N^u}\log\Big(\lambda_{e^u_k}(\boldsymbol{x_u}) \\ & + \sum_{j=1}^{k-1}\sum_{d=1}^{D} a^d_{e^u_k,e^u_j} g_d(t^u_k - t^u_j)\Big)\Big\} + \alpha_1\Big(\sum_{d=1}^{D}\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(a^d_{i,j})^2}{2}\Big) + \alpha_2\Big(\sum_{d=1}^{D}\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(a^d_{i,j} + a^d_{j,i})^2}{2}\Big),\end{aligned}$$

and the learning problem is:

$$\min_{\Theta} F(\Theta). \tag{2.14}$$

2.4.2 Parameters estimation

In order to estimate parameters, we use an Expectation-Maximization (EM) algorithm. It consists in introducing the notion of complete likelihood, which assumes that we could observe, for each event, which previous event has triggered this event (see [2]).

To do so, we construct a surrogate objective function (expectation step), whose iterative minimization allows initial objective function to decrease monotonically (maximization step). We denote $\hat{\Theta}^{(l)}$ the estimation of parameters at iteration l: $\hat{\Theta}^{(l)} = (\hat{\boldsymbol{\mu}}^{(l)}, \hat{\boldsymbol{A}}^{(l)})$. The surrogate objective function denoted $F_s(\Theta, \hat{\Theta}^{(l)})$ equals:

$$F_s(\Theta, \hat{\Theta}^{(l)}) = -Q(\Theta, \hat{\Theta}^{(l)}) + R_1(\Theta) + \hat{R}_2^{(l)}(\Theta, \hat{\Theta}^{(l)}), \tag{2.15}$$

where:

1.
$$\begin{aligned} Q(\Theta, \hat{\Theta}^{(l)}) = & \sum_{u=1}^{U} \Big\{ -\sum_{i=1}^{m} \Big(T_u \lambda_i(\boldsymbol{x_u}) + \sum_{k=1}^{N^u(T_u)} \sum_{d=1}^{D} a^d_{i,e^u_k} G_d(T_u - t^u_k) \Big) \\ & + \sum_{k=1}^{N^u(T_u)} p^{(l)}_{e^u_k, e^u_k} \log\Big(\frac{\lambda_{e^u_k}(\boldsymbol{x_u})}{p^{(l)}_{e^u_k, e^u_k}} \Big) + \sum_{j=1}^{k-1} \sum_{d=1}^{D} p^{(l)}_{e^u_k, e^u_j} \log\Big(\frac{a^d_{e^u_k, e^u_j} g_d(t^u_k - t^u_j)}{p^{(l)}_{e^u_k, e^u_j}} \Big) \Big\}; \end{aligned} \tag{2.16}$$

2.
$$p^{(l)}_{e^u_k, e^u_k} = \frac{\hat{\mu}^{(l)}_{e^u_k} f_{e^u_k}(\boldsymbol{x_u})}{\hat{\mu}^{(l)}_{e^u_k} f_{e^u_k}(\boldsymbol{x_u}) + \sum_{j=1}^{k-1} \sum_{d=1}^{D} \hat{a}^{d,(l)}_{e^u_k, e^u_j} g_d(t^u_k - t^u_j)}, \tag{2.17}$$

which represents the probability that event (e^u_k, t^u_k) was triggered by the background intensity of event of type e^u_k;

3.
$$p^{(l)}_{e^u_k, e^u_j} = \frac{\sum_{d=1}^{D} \hat{a}^{d,(l)}_{e^u_k, e^u_j} g_d(t^u_k - t^u_j)}{\hat{\mu}^{(l)}_{e^u_k} f_{e^u_k}(\boldsymbol{x_u}) + \sum_{j=1}^{k-1} \sum_{d=1}^{D} \hat{a}^{d,(l)}_{e^u_k, e^u_j} g_d(t^u_k - t^u_j)}, \tag{2.18}$$

which represents the probability that event (e^u_k, t^u_k) was triggered by event (e^u_j, t^u_j);

4.
$$\hat{R}_2^{(l)}(\Theta, \hat{\Theta}^{(l)}) = \alpha_2 \Big(\sum_{d=1}^{D} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{(a^d_{i,j} + \hat{a}^{d,(l)}_{j,i})^2}{2} \Big). \tag{2.19}$$

We update iteratively the parameters estimation by calculating:

$$\Theta^{(l+1)} = \underset{\Theta}{\text{argmax}} F_s(\Theta, \hat{\Theta}^{(l)}). \tag{2.20}$$

We solve $\frac{\partial F}{\partial \Theta} = 0$ (see Appendix) and we obtain, for all $d \in \{1, ..., D\}, k \in \{1, ..., m\}, (i, j) \in \{1, ..., m\}^2$:

$$\hat{\mu}_k^{(l+1)} = \frac{\sum_{u=1}^{U} \sum_{e_i^u = k} p^{(l)}_{e_i^u, e_i^u}}{\sum_{u=1}^{U} T_u}; \tag{2.21}$$

$$\hat{a}_{i,j}^{d,(l+1)} = \frac{-(A_{i,j} + \alpha_2 \hat{a}_{j,i}^{d,(l)}) + \sqrt{(A_{i,j} + \alpha_2 \hat{a}_{j,i}^{d,(l)})^2 - 4(\alpha_1 + \alpha_2) B_{i,j}^{(l)}}}{2(\alpha_1 + \alpha_2)}, \tag{2.22}$$

where

$$A_{i,j} = -\sum_{u=1}^{U} \sum_{e_k^u = j} G_d(T_u - t_k^u), \tag{2.23}$$

$$B_{i,j}^{(l)} = \sum_{u=1}^{U} \sum_{e_k^u = i} \sum_{e_m^u = j} p^{(l)}_{e_k^u, e_m^u}. \tag{2.24}$$

Optimizing F_s this way allows us to decrease the Negative Log-Likelihood monotonically. Indeed, from the two following inequalities (see a proof from Appendix in [81]):

- $-Q(\Theta, \hat{\Theta}^{(l)}) \geq \text{NLL}(\Theta)$;
- $-Q(\hat{\Theta}^{(l)}, \hat{\Theta}^{(l)}) = \text{NLL}(\hat{\Theta}^{(l)})$,

we could deduce that $\text{NLL}(\hat{\Theta}^{(l-1)}) = -Q(\hat{\Theta}^{(l-1)}, \hat{\Theta}^{(l-1)}) \geq -Q(\hat{\Theta}^{(l)}, \hat{\Theta}^{(l-1)}) \geq \text{NLL}(\hat{\Theta}^{(l)})$. We update iteratively $\hat{\Theta}^{(l)} = (\hat{\boldsymbol{\mu}}^{(l)}, \hat{\boldsymbol{A}}^{(l)})$ while the relative difference between $\text{NLL}(\hat{\Theta}^{(l-1)})$ and $\text{NLL}(\hat{\Theta}^{(l)})$, denoted:

$$\text{Err}(l) = \left| \frac{\text{NLL}(\hat{\Theta}^{(l)})}{\text{NLL}(\hat{\Theta}^{(l-1)})} - 1 \right|, \tag{2.25}$$

is above ε and the number of iterations is below N_{iter}. This methodology is summarized in Algorithm 5.

Algorithm 5: Parameters estimation

Result: Estimation of $\Theta = (\boldsymbol{\mu}, \boldsymbol{A})$

Initialize $l = 0$ and $(\hat{\boldsymbol{\mu}}^{(l)}, \hat{\boldsymbol{A}}^{(l)})$;

$l = l + 1$;

Update $(\hat{\boldsymbol{\mu}}^{(l)}, \hat{\boldsymbol{A}}^{(l)})$ according to Equations (2.21) and (2.22);

while *Err*$(l) \geq \varepsilon$ *and* $l \leq N_{iter}$ **do**

  $l = l + 1$;

  Update $(\hat{\boldsymbol{\mu}}^{(l)}, \hat{\boldsymbol{A}}^{(l)})$ according to Equations (2.21) and (2.22);

end

2.5 Mathematical properties

In this section, we are going to explicit several mathematical properties of the Γ-Hawkes processes (defined by (2.3)): we calculate the expectation and the variance for any time $t \geq 0$ and we state a central limit theorem.

Assumption 3 *For the whole Section 2.5, we consider a* Γ*-Hawkes process of conditional intensity function:*

$$\lambda^*(t) = \lambda + \int_0^t \mu(t-u)dN(u) = \lambda + \alpha \sum_{k=1}^{N(t)} \frac{(t-t_k)^{k_1-1}\exp(\frac{-(t-t_k)}{k_2})}{k_2^{k_1}\Gamma(k_1)}, \tag{2.26}$$

where $\alpha \in \mathbb{R}_+^*, k_1, k_2 \in \mathbb{R}_+^*$*,* $N(t) \geq 0$ *(number of events that occurred between* 0 *and* t*),* t_k*,* $k \in \{1, ..., N(t)\}$*. We are going to explicit results for* $k_1 = 1$ *(i.e. exponential case) and* $k_1 = 2$*.*

This study aims to understand the behaviour of the Hawkes processes with respect to the parameters introduced previously.

2.5.1 Behaviour with respect to the values of α

In this section, we are going to see that the Hawkes process dynamics depend on values of α (defined in Assumption 3). To do so, let us focus on $g(t) = \mathbb{E}[\lambda^*(t)]$, which will be used to calculate the expectation of the Hawkes process later.
We could show that g follows a renewal equation:

$$g(t) = \lambda + \int_0^t \mu(t-u)g(u)du. \tag{2.27}$$

According to [82], the behaviour of $g(t)$, when t goes to infinity, depends on the value of:

$$\int_0^{+\infty} \mu(u)du = \int_0^{+\infty} \alpha \frac{\beta^{k_1} u^{k_1-1}\exp(-\beta u)}{\Gamma(k_1)} du = \alpha. \tag{2.28}$$

We distinguish, in the classic literature about point processes:

- the defective case: $\alpha < 1$;
- the proper case: $\alpha = 1$;
- the excessive case: $\alpha > 1$.

The Hawkes process intensity goes to infinity in the excessive case, leading to an explosion of the counting process. This situation does not suit to any real event we would like to model. The proper case is not studied here, because it does not correspond to a real case. We concentrate our attention in the defective case. In this case, g admits a finite limit in $+\infty$, which will be explicited later.

Assumption 4 *For the rest of the Section 2.5, we consider a Hawkes process of conditional intensity function described in Assumption 3, with* $0 < \alpha < 1$ *(defective case).*

2.5.2 Expectation evaluation

The first indicator which is relevant for our study is the evaluation of the average number of events through time. We propose a calculation method which could be applied for any value of $k_1 \in \mathbb{N}^*$. We will develop the calculations for three values of the shape parameter: $k_1 = 1$ (exponential triggering function), $k_1 = 2$ and $k_1 = 3$. Since a Hawkes process is the sum of a background part of intensity λ and an self-exciting part, we expect that the expectation is higher than the one of a Poisson process of parameter λ, which is λt for all time $t > 0$.

Proposition 3 *The expectation of the* Γ*-Hawkes process (i.e. conditional intensity function given by Equation (2.26)) is:*

1. *For* $k_1 = 1$ *(i.e.* $\mu(t) = \alpha \frac{\exp(-\frac{t}{k_2})}{k_2}$*, exponential decay):*

$$\mathbb{E}(N(t)) = \frac{\lambda}{1-\alpha} t - \frac{\alpha \lambda k_2}{(1-\alpha)^2} \Big[1 - \exp\Big(- \frac{1-\alpha}{k_2} t \Big) \Big]. \tag{2.29}$$

2. *For* $k_1 = 2$ *(i.e.* $\mu(t) = \alpha \frac{t \exp(-\frac{t}{k_2})}{k_2^2}$*):*

$$\begin{aligned} \mathbb{E}(N(t)) = \frac{\lambda}{1-\alpha} t &+ \frac{\lambda \sqrt{\alpha} k_2}{2(1+\sqrt{\alpha})^2} \Big[1 - \exp\Big(- \frac{1+\sqrt{\alpha}}{k_2} t \Big) \Big] \\ &- \frac{\lambda \sqrt{\alpha} k_2}{2(1-\sqrt{\alpha})^2} \Big[1 - \exp\Big(- \frac{1-\sqrt{\alpha}}{k_2} t \Big) \Big]. \end{aligned} \tag{2.30}$$

3. *For* $k_1 = 3$ *(i.e.* $\mu(t) = \alpha \frac{t^2 \exp(-\frac{t}{k_2})}{k_2^3 \Gamma(3)}$*):*

$$\begin{aligned} \mathbb{E}(N(t)) = \frac{\lambda}{1-\alpha} t + \frac{\lambda \alpha B}{k_2^3} \exp(-r_1 t) &+ \frac{\lambda \alpha C}{k_2^3} \cos(\omega t) \exp(-a t) \\ &+ \frac{\lambda \alpha (D - C a)}{k_2^3 \omega} \sin(\omega t) \exp(-a t), \end{aligned} \tag{2.31}$$

where

$$\begin{aligned} B &= \frac{1}{r_1(r_1 a_1 - r_1^2 - a_2)}, \quad C = \frac{r_1 - a_1}{a_2(r_1 a_1 - r_1^2 - a_2)}, \quad D = \frac{r_1 a_1 - a_1^2 + a_2}{a_2(r_1 a_1 - r_1^2 - a_2)}, \\ r_1 &= \frac{1 - \alpha^{1/3}}{k_2}, \quad a_1 = \frac{(2 + \alpha^{1/3})}{k_2}, \quad a_2 = \frac{(1-\alpha)}{(1-\alpha^{1/3}) k_2^2}, \\ a &= \frac{(2 + \alpha^{1/3})}{2 k_2}, \quad \omega = \frac{\sqrt{3} \alpha^{1/3}}{2 k_2}. \end{aligned}$$

See Section 2.6.2 for the proof, which is based on Laplace transform of $g(t) = \mathbb{E}(\lambda^*(t))$ and could be applied for any value of k_1. We could notice that despite the formula complexity increases with k_1, we could observe several common trends. We first notice that for $\alpha \to 0$ (null triggering function), we obtain $\mathbb{E}(N(t)) \underset{\alpha\to 0}{\longrightarrow} \lambda t$. This result is natural since we already mentioned that a Hawkes process with a null triggering function is equivalent to a Poisson process of parameter λ. For every value of the shape parameter, the expectation reveals three regimes for the Γ-Hawkes process. As an example, we represent the expectation for $k_1 = 1, k_2 = 9, \alpha = 0.9, \lambda = 0.3$, which follows equation (2.29), in Figure 2. It contains four plots:

- Top-left: expectation from equation (2.29) in blue and the so-called stationary regime $t \to \frac{\lambda}{1-\alpha} t$ in red, from $t = 0$ to 3000;
- Bottom-left: the relative difference between the expectation and the stationary regime on this time period;
- Top-right: focus on the expectation from $t = 0$ to 500;
- Bottom-right: plot of the exponential term from (2.29), $t \to \exp\left(-\frac{1-\alpha}{k_2} t\right)$.

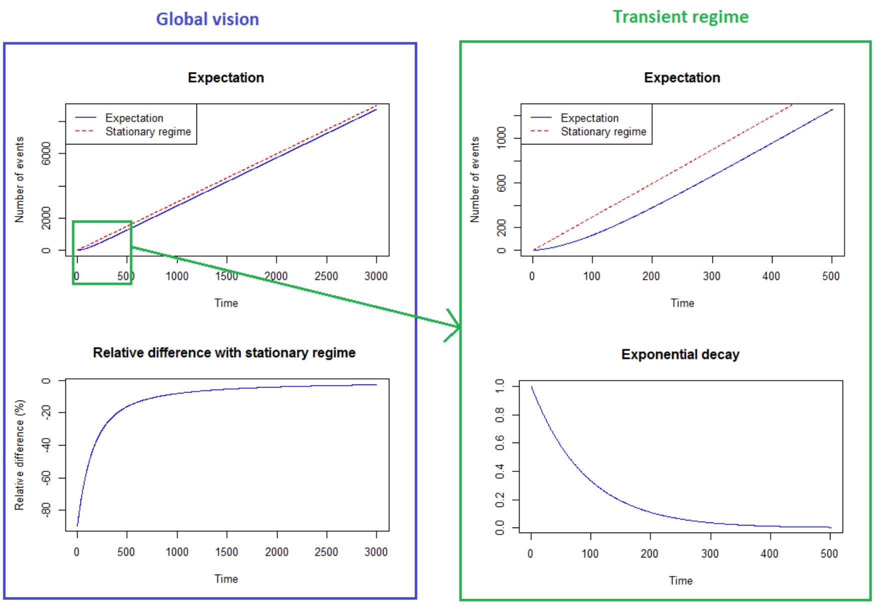


Figure 2.6: Expectation for $k_1 = 1, k_2 = 9, \alpha = 0.9, \lambda = 0.3$

We distinguish three regimes in this example:

- The transient regime (from $t = 0$ to 500 approximately): it is the time period necessary for the exponential term in (2.29) to become null;
- The intermediary regime (from $t = 500$ to 2000 approximately): the exponential term is null but the constant term could not be neglected before the stationary regime. On this period, the relative difference between the expectation and the stationary regime is above 5%;

- The stationary regime: for any value of the parameter scale, and for any Hawkes process (not only for Γ-Hawkes processes), $\mathbb{E}(N(t)) \underset{t\to+\infty}{\sim} \frac{\lambda}{1-\alpha}t$ (see [83]). The constant term in (2.29) becomes negligible before the linear term. After the first two regimes, the average number of occurrences of the Hawkes process is the same as for a Poisson process of parameter $\frac{\lambda}{1-\alpha}$.

We conclude in this discussion that the expectation could not be approximated correctly by the stationary regime from $t = 0$ to 2000.

2.5.3 Variance

We study now the variance of the Hawkes process, in order to see how the occurrences are spread out from the expectation calculated previously. We define $\widetilde{N}(\cdot)$ as the stationary Hawkes process.

Proposition 4 *The variance of the* Γ*-Hawkes process (i.e. the conditional intensity function is given by equation (2.26)) is:*

1. *For* $k_1 = 1$ *(i.e.* $\mu(t) = \alpha\frac{\exp(-\frac{t}{k_2})}{k_2}$*):*

$$\mathbb{V}(\widetilde{N}(t)) = \frac{\lambda}{(1-\alpha)^3}t - \frac{\lambda\alpha(2-\alpha)k_2}{(1-\alpha)^4}\Big[1 - \exp\Big(-\frac{1-\alpha}{k_2}t\Big)\Big]. \tag{2.32}$$

2. *For* $k_1 = 2$ *(i.e.* $\mu(t) = \alpha\frac{t\exp(-\frac{t}{k_2})}{k_2^2}$*):*

$$\mathbb{V}(\widetilde{N}(t)) = C_1 + C_2 t + C_3\exp(-\omega_1 t) + C_4\exp(-\omega_2 t), \text{ where:} \tag{2.33}$$

$$\omega_1 = \frac{1+\sqrt{\alpha}}{k_2}, \tag{2.34}$$

$$\omega_2 = \frac{1-\sqrt{\alpha}}{k_2}, \tag{2.35}$$

$$C_1 = \frac{-\alpha k_2\lambda[8-5\alpha+\alpha^2]}{2(1-\alpha)^4}, \tag{2.36}$$

$$C_2 = \frac{\lambda}{(1-\alpha)^3}, \tag{2.37}$$

$$C_3 = \frac{-\sqrt{\alpha}\lambda k_2[4-3\alpha-\sqrt{\alpha}\alpha]}{4(1-\alpha)^2(1+\sqrt{\alpha})^2}, \tag{2.38}$$

$$C_4 = \frac{\sqrt{\alpha}\lambda k_2[4-3\alpha+\sqrt{\alpha}\alpha]}{4(1-\alpha)^2(1-\sqrt{\alpha})^2}. \tag{2.39}$$

See Section 2.6.3 for the proof, in which we use the classic assumption that covariance density does not depend on time, assumption valid only if the process is stationary. However, we remark that theoretical variance is close to the results from simulations, see Section

4.5 for a comparison between simulated and actual variance. A similar comparison in [70] shows that theoretical variance based on stationary assumptions is a fine approximation as well. Therefore, we consider that $\mathbb{V}(N(t))$ approximately equals $\mathbb{V}(\widetilde{N}(t))$ from now on.

Let us observe the behaviour of the variance. We first notice that for $\alpha \to 0$ (null triggering function), we obtain $\mathbb{V}(N(t)) \underset{\alpha \to 0}{\longrightarrow} \lambda t$. This result is natural since we already mentioned that a Hawkes process with a null triggering function is equivalent to a Poisson process of parameter λ. Secondly, we remark that $\mathbb{V}(N(t)) \underset{t \to +\infty}{\sim} \frac{\lambda}{(1-\alpha)^3} t$. We have seen previously that $\mathbb{E}(N(t)) \underset{t \to +\infty}{\sim} \frac{\lambda}{1-\alpha} t$: it is logical that we obtain as a limit a variance greater than $\frac{\lambda}{1-\alpha} t$ since a Hawkes process is more volatile than a Poisson process. The limit of the variance is calculated in Proposition 3 from [83], which gives the same result. It is also possible to do an analysis in terms of regimes similar to the expectation.

2.5.4 Central limit theorem for Hawkes processes

From Theorem 2 of [84], we state a central limit theorem for Hawkes process, when $t \to +\infty$. The following proposition is true for any $k_1 \geq 1$:

Proposition 5 ***(Central limit theorem for Hawkes processes)*** *We have the following convergence in distribution:*

$$\lim_{t \to +\infty} \frac{N(t) - \bar{\lambda} t}{\sqrt{t}} \overset{d}{=} \sigma \mathcal{N}(0, 1), \tag{2.40}$$

where:

- $\bar{\lambda} = \frac{\lambda}{1-\alpha}$;
- $\sigma^2 = \frac{\lambda}{(1-\alpha)^3}$;
- $\overset{d}{=}$ *means equality in distribution.*

We recognize in this central limit theorem the limits of expectation and variance calculated previously. The central limit theorem gives the distribution of the number of events at a large time horizon.

2.6 Proofs

2.6.1 Proof of equations (2.21) and (2.22)

We update $\hat{\Theta}^{(l)}$ such that $\hat{\Theta}^{(l+1)} = \underset{\Theta}{\text{argmax}} F_s(\Theta, \hat{\Theta}^{(l)})$. We solve $\frac{\partial F_s(\Theta, \hat{\Theta}^{(l)})}{\partial \Theta} = 0$.

$$\begin{aligned}
F_s(\Theta, \hat{\Theta}^{(l)}) = & -Q(\Theta, \hat{\Theta}^{(l)}) + R_1(\Theta) + \hat{R}_2^{(l)}(\Theta, \hat{\Theta}^{(l)}) \\
= & \sum_{u=1}^{U} \Big\{ -\sum_{i=1}^{m} \Big(T_u \lambda_i(\boldsymbol{x_u}) + \sum_{k=1}^{N^u(T_u)} \sum_{d=1}^{D} a^d_{i,e^u_k} G_d(T_u - t^u_k) \Big) \\
+ & \sum_{k=1}^{N^u(T_u)} p^{(l)}_{e^u_k, e^u_k} \log\Big(\frac{\lambda_{e^u_k}(\boldsymbol{x_u})}{p^{(l)}_{e^u_k, e^u_k}} \Big) + \sum_{j=1}^{k-1} \sum_{d=1}^{D} p^{(l)}_{e^u_k, e^u_j} \log\Big(\frac{a^d_{e^u_k, e^u_j} g_d(t^u_k - t^u_j)}{p^{(l)}_{e^u_k, e^u_j}} \Big) \\
+ & \alpha_1 \Big(\sum_{d=1}^{D} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{(a^d_{i,j})^2}{2} \Big) + \alpha_2 \Big(\sum_{d=1}^{D} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{(a^d_{i,j} + \hat{a}^{d,(l)}_{j,i})^2}{2} \Big) \Big\}.
\end{aligned}$$

For all $k \in \{1, ..., m\}$,

$$\frac{\partial F_s(\Theta, \hat{\Theta}^{(l)})}{\partial \mu_k} = -\sum_{u=1}^{U} T_u + \Big(\sum_{u=1}^{U} \sum_{e^u_i = k} p^{(l)}_{e^u_i, e^u_i} \Big) \frac{1}{\mu_k},$$

therefore we set $\hat{\mu}_k^{(l+1)}$ such that $\left(\frac{\partial F_s}{\partial \mu_k} \right)_{\mu_k = \hat{\mu}_k^{(l+1)}} = 0$, which leads to:

$$\hat{\mu}_k^{(l+1)} = \frac{\sum_{u=1}^{U} \sum_{e^u_i = k} p^{(l)}_{e^u_i, e^u_i}}{\sum_{u=1}^{U} T_u}.$$

For all $d \in \{1, ..., D\}, (i, j) \in \{1, ..., m\}^2$,

$$\frac{\partial F_s(\Theta, \hat{\Theta}^{(l)})}{\partial a^d_{i,j}} = -\sum_{u=1}^{U} \sum_{e^u_k = j} G_d(T_u - t^u_k) + \Big(\sum_{u=1}^{U} \sum_{e^u_k = i} \sum_{e^u_m = j} p^{(l)}_{e^u_k, e^u_k} \Big) \frac{1}{a^d_{i,j}} + (\alpha_1 + \alpha_2) a^d_{i,j} + \alpha_2 \hat{a}^{d,(l)}_{j,i}.$$

We denote:

$$A_{i,j} = -\sum_{u=1}^{U} \sum_{e^u_k = j} G_d(T_u - t^u_k) \text{ and}$$

$$B^{(l)}_{i,j} = \sum_{u=1}^{U} \sum_{e^u_k = i} \sum_{e^u_m = j} p^{(l)}_{e^u_k, e^u_m}.$$

We set $\hat{a}^{d,(l+1)}_{i,j}$ such that $\left(\frac{\partial F_s}{\partial a^d_{i,j}} \right)_{a^d_{i,j} = \hat{a}^{d,(l+1)}_{i,j}} = 0$, which leads to:

$$\begin{aligned}
A_{i,j} + B^{(l)}_{i,j} \frac{1}{a^d_{i,j}} + (\alpha_1 + \alpha_2) a^d_{i,j} + \alpha_2 \hat{a}^{d,(l)}_{j,i} &= 0, \\
(\alpha_1 + \alpha_2)(a^d_{i,j})^2 + (A_{i,j} + \alpha_2 \hat{a}^{d,(l)}_{j,i}) a^d_{i,j} + B^{(l)}_{i,j} &= 0,
\end{aligned}$$

$$\hat{a}_{i,j}^{d,(l+1)} = \frac{-(A_{i,j} + \alpha_2 \hat{a}_{j,i}^{d,(l)}) + \sqrt{(A_{i,j} + \alpha_2 \hat{a}_{j,i}^{d,(l)})^2 - 4(\alpha_1 + \alpha_2) B_{i,j}^{(l)}}}{2(\alpha_1 + \alpha_2)}.$$

□

2.6.2 Proof of Proposition 3

The proof of Proposition 3 is based on a Laplace transform, since the calculation of $\mathbb{E}(N(t))$ implies a convolution product of two functions.

1. Laplace transform of $g(t) = \mathbb{E}[\lambda^*(t)]$

   Since $\mathbb{E}(N(t)) = \int_0^t \mathbb{E}[\lambda^*(t)]dt$, we first focus on $\mathbb{E}[\lambda^*(t)]$, that we denote $g(t)$. We saw that:
   $$g(t) = \lambda + \int_0^t \mu(t-u)g(u)du. \tag{2.41}$$

   We recognize a convolution product of g and μ, denoted $g * \mu$. Therefore it is much easier to work into Laplace domain, since $\mathcal{L}\Big[g * \mu(t)\Big](s) = \mathcal{L}\Big[g(t)\Big](s) \times \mathcal{L}\Big[\mu(t)\Big](s)$. By transforming (2.41) into Laplace domain:

   $$\begin{aligned} \mathcal{L}\Big[g(t)\Big](s) &= \mathcal{L}\Big[\lambda + \int_0^t \mu(t-u)g(u)du\Big](s) \\ &= \mathcal{L}\Big[\lambda\Big](s) + \mathcal{L}\Big[\mu(t)\Big](s) \times \mathcal{L}\Big[g(t)\Big](s). \end{aligned}$$

   By rearranging, we obtain:

   $$\mathcal{L}\Big[g(t)\Big](s) = \frac{\mathcal{L}\Big[\lambda\Big](s)}{1 - \mathcal{L}\Big[\mu(t)\Big](s)}. \tag{2.42}$$

2. Inverse of $g(t)$ Laplace transform

   We turn the Laplace transform into the temporal domain. In our study, μ is a Gamma density function and is given by equation (2.3). We can show that the Laplace transform is (see [85]):

   $$\mathcal{L}\Big[\mu(t)\Big](s) = \frac{\alpha}{(1 + k_2 s)^{k_1}}. \tag{2.43}$$

By using (2.42) in (2.43), we obtain:

$$\mathcal{L}\Big[g(t)\Big](s) = \frac{\mathcal{L}\Big[\lambda\Big](s)}{1-\mathcal{L}\Big[\mu(t)\Big](s)} = \frac{\lambda}{s\Big(1-\frac{\alpha}{(1+k_2 s)^{k_1}}\Big)}.$$

The complexity of this Laplace transform increases with the value of k_1. In particular, from $k_1 = 3$, more calculation is needed to identify the right inverse Laplace transform.

- For $k_1 = 1$ (see (2.5)):

$$\mathcal{L}\Big[g(t)\Big](s) = \frac{\lambda}{s\Big(1-\frac{\alpha}{(1+k_2 s)}\Big)} = \frac{\Big(\frac{\lambda}{1-\alpha}\Big)}{s} - \frac{\Big(\frac{\alpha\lambda}{1-\alpha}\Big)}{\frac{1-\alpha}{k_2}+s}. \tag{2.44}$$

  By transforming (2.44) into the temporal domain, we obtain:

$$g(t) = \mathbb{E}(\lambda^*(t)) = \frac{\lambda}{1-\alpha} - \frac{\alpha\lambda}{1-\alpha}\exp\Big(-\frac{1-\alpha}{k_2}t\Big).$$

- For $k_1 = 2$ (see 2.4):

$$\begin{aligned}\mathcal{L}\Big[g(t)\Big](s) = {} & \frac{\lambda}{k_2^2 s\Big(\frac{1+\sqrt{\alpha}}{k_2}+s\Big)\Big(\frac{1-\sqrt{\alpha}}{k_2}+s\Big)} + \frac{2\lambda}{k_2\Big(\frac{1+\sqrt{\alpha}}{k_2}+s\Big)\Big(\frac{1-\sqrt{\alpha}}{k_2}+s\Big)} \\ & + \frac{\lambda s}{\Big(\frac{1+\sqrt{\alpha}}{k_2}+s\Big)\Big(\frac{1-\sqrt{\alpha}}{k_2}+s\Big)}.\end{aligned} \tag{2.45}$$

  By turning (2.45) into the temporal domain, we obtain:

$$\begin{aligned} g(t) = \mathbb{E}(\lambda(t)) = \frac{\lambda}{1-\alpha} & + \frac{\lambda\sqrt{\alpha}}{2(1+\sqrt{\alpha})}\exp\Big(-\frac{1+\sqrt{\alpha}}{k_2}t\Big) \\ & -\frac{\lambda\sqrt{\alpha}}{2(1-\sqrt{\alpha})}\exp\Big(-\frac{1-\sqrt{\alpha}}{k_2}t\Big).\end{aligned}$$

- For $k_1 = 3$ (see 2.4):

$$\begin{aligned}\mathcal{L}\Big[g(t)\Big](s) = {} & \frac{\lambda}{s\Big(1-\frac{\alpha}{(1+k_2 s)^3}\Big)} \\ = {} & \frac{\lambda}{s} + \frac{\lambda\alpha}{s((1+k_2 s)^3-\alpha)} \\ = {} & \frac{\lambda}{s} + \frac{\lambda\alpha}{k_2^3 s(s+\frac{1-\alpha^{1/3}}{k_2})(s^2+a_1 s+a_2)} \\ = {} & \frac{\lambda}{s} + \frac{\lambda\alpha}{k_2^3}\Big[\frac{A}{s} + \frac{B}{s+r_1} + \frac{Cs+D}{(s+a)^2+\omega^2}\Big],\end{aligned}$$

by partial fraction decomposition, where:
$r_1 = \frac{1-\alpha^{1/3}}{k_2}, a_1 = \frac{(2+\alpha^{1/3})}{k_2}, a_2 = \frac{(1-\alpha)}{(1-\alpha^{1/3})k_2^2}, a = \frac{(2+\alpha^{1/3})}{2k_2}, \omega = \frac{\sqrt{3}\alpha^{1/3}}{2k_2}$.

From the above partial fraction decomposition, (A, B, C, D) is the solution of the following linear system:

$$\begin{cases} A + B + C = 0 \\ (a_1 + r_1)\, A + a_1\, B + r_1\, C + D = 0 \\ (a_2 + r_1 a_1)\, A + a_2\, B + r_1\, D = 0 \\ r_1 a_2\, A = 1 \end{cases}$$

The solution is:

$$\begin{aligned} A &= \frac{1}{r_1 a_2}, \\ B &= \frac{1}{r_1(r_1 a_1 - r_1^2 - a_2)}, \\ C &= \frac{r_1 - a_1}{a_2(r_1 a_1 - r_1^2 - a_2)}, \\ D &= \frac{r_1 a_1 - a_1^2 + a_2}{a_2(r_1 a_1 - r_1^2 - a_2)}. \end{aligned}$$

Therefore the Laplace transform equals:

$$\begin{aligned} \mathcal{L}\big[g(t)\big](s) &= \frac{\lambda}{s} + \frac{\lambda\alpha}{k_2^3}\Big[\frac{A}{s} + \frac{B}{s + r_1} + \frac{Cs + D}{(s+a)^2 + \omega^2}\Big] \\ &= \frac{\lambda}{s} + \frac{\lambda\alpha}{k_2^3}\Big[\frac{A}{s} + \frac{B}{s + r_1} + \frac{C(s+a)}{(s+a)^2 + \omega^2} + \frac{\frac{(D-Ca)}{\omega}\omega}{(s+a)^2 + \omega^2}\Big] \end{aligned}$$

$$= \frac{\lambda}{(1-\alpha)s} + \frac{\lambda\alpha B}{k_2^3(s + r_1)} + \frac{\lambda\alpha C}{k_2^3}\frac{(s+a)}{(s+a)^2 + \omega^2} + \frac{\lambda\alpha(D - Ca)}{k_2^3\omega}\frac{\omega}{(s+a)^2 + \omega^2}. \tag{2.46}$$

By turning (2.46) into the temporal domain, we obtain:

$$\begin{aligned} g(t) = \mathbb{E}(\lambda(t)) = \frac{\lambda}{1-\alpha} &+ \frac{\lambda\alpha B}{k_2^3}\exp(-r_1 t) + \frac{\lambda\alpha C}{k_2^3}\cos(\omega t)\exp(-at) \\ &+ \frac{\lambda\alpha(D - Ca)}{k_2^3\omega}\sin(\omega t)\exp(-at). \end{aligned}$$

3. Calculation of $\mathbb{E}(N(t))$ from $g(t)$

From $g(t)$, we calculate $\mathbb{E}(N(t))$:

$$\mathbb{E}(N(t)) = \int_0^t g(t)dt,$$

which leads us to (2.29), (2.30) and (2.31).

□

2.6.3 Proof of Proposition 4

Proof inspired from Appendix A.2 of [67].

1. Link between the variance and the covariance density $\Phi(\tau)$

   We calculate the variance from the so-called covariance density $\Phi(\tau)$, $\tau > 0$, defined in [66]. According to [66], the covariance density is given by the following equation:

$$\begin{aligned}\Phi(\tau) = & \ \bar{\lambda}\mu(\tau) + \int_{-\infty}^{\tau} \mu(\tau - u)\Phi(u)du \\ = & \ \bar{\lambda}\mu(\tau) + \int_0^{+\infty} \mu(\tau + u)\Phi(u)du + \int_0^{\tau} \mu(\tau - u)\Phi(u)du, \end{aligned} \tag{2.47}$$

   where $\bar{\lambda} = \frac{\lambda}{1-\alpha}$. Considering the stationary Hawkes process $\widetilde{N}(\cdot)$, the variance $\mathbb{V}(\widetilde{N}(t))$ and the covariance density are linked by the following equation:

$$\mathbb{V}(\widetilde{N}(t)) = \bar{\lambda}t + 2\int_0^t \int_0^{t_1} \Phi(t_2 - t_1)dt_1dt_2. \tag{2.48}$$

2. Laplace transform of $\Phi(\tau)$

   We first calculate $\Phi(\tau)$. We recognize a convolution product in equation (2.47). Thus we work in the Laplace domain:

$$\begin{aligned}\mathcal{L}\Big[\Phi(\tau)\Big](s) &= \mathcal{L}\Big[\bar{\lambda}\mu(\tau) + \int_0^{+\infty} \mu(\tau + u)\Phi(u)du + \int_0^{\tau} \mu(\tau - u)\Phi(u)du\Big](s) \\ &= \bar{\lambda}\mathcal{L}\Big[\mu(\tau)\Big](s) + \underbrace{\mathcal{L}\Big[\int_0^{+\infty} \mu(\tau + u)\Phi(u)du\Big](s)}_{=:l_1(s)} + \mathcal{L}\Big[\Phi(\tau)\Big](s) \times \mathcal{L}\Big[\mu(\tau)\Big](s). \end{aligned} \tag{2.49}$$

   For $k_1 = 2$:

$$
\begin{aligned}
l_1(s) = & \ \mathcal{L}\Big[\int_0^{+\infty} \mu(\tau+u)\Phi(u)du\Big](s) \\
= & \ \mathcal{L}\Big[\int_0^{+\infty} \frac{\alpha(\tau+u)\exp(-\frac{\tau+u}{k_2})}{k_2^2}\Phi(u)du\Big](s) \\
= & \ \mathcal{L}\Big[\int_0^{+\infty} \frac{\alpha\tau\exp(-\frac{\tau}{k_2})\exp(-\frac{u}{k_2})}{k_2^2}\Phi(u)du\Big](s) \\
+ & \ \mathcal{L}\Big[\int_0^{+\infty} \frac{\alpha u\exp(-\frac{\tau}{k_2})\exp(-\frac{u}{k_2})}{k_2^2}\Phi(u)du\Big](s).
\end{aligned}
$$

By using the definition of the Laplace transform:

$$
\begin{aligned}
l_1(s) = & \ \frac{\alpha}{k_2^2}\int_0^{+\infty} \tau\exp(-(s+\frac{1}{k_2})\tau)\underbrace{\Big[\int_0^{+\infty} \exp(-\frac{u}{k_2})\Phi(u)du\Big]}_{\mathcal{L}\left[\Phi(\tau)\right](\frac{1}{k_2})} d\tau \\
+ & \ \frac{\alpha}{k_2^2}\int_0^{+\infty} \exp(-(s+\frac{1}{k_2})\tau)\underbrace{\Big[\int_0^{+\infty} u\exp(-\frac{u}{k_2})\Phi(u)du\Big]}_{-\mathcal{L}\left[\Phi(\tau)\right]'(\frac{1}{k_2})} d\tau \\
= & \ \frac{\alpha}{k_2^2(s+\frac{1}{k_2})}\Big[\frac{1}{(s+\frac{1}{k_2})}\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) - \mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})\Big].
\end{aligned}
$$

By replacing $l_1(s)$ in (2.49) we obtain:

$$
\begin{aligned}
\mathcal{L}\Big[\Phi(\tau)\Big](s) = & \ \bar{\lambda}\mathcal{L}\Big[\mu(\tau)\Big](s) + \frac{\alpha}{k_2^2(s+\frac{1}{k_2})}\Big[\frac{1}{(s+\frac{1}{k_2})}\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) - \mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})\Big] \\
+ & \ \mathcal{L}\Big[\Phi(\tau)\Big](s)\times\mathcal{L}\Big[\mu(\tau)\Big](s). \qquad (2.50)
\end{aligned}
$$

To solve (2.50) and find Φ, we need to calculate $\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2})$ and $\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})$.

First, by substituting $s = \frac{1}{k_2}$ in (2.50), rearranging and using (2.43) for $\mathcal{L}\Big[\mu(\tau)\Big](s)$, we obtain:

$$
\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2}) + \frac{(2-\alpha)k_2}{\alpha}\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) = \frac{\bar{\lambda}k_2}{2}. \qquad (2.51)
$$

Second, by differentiating (2.50), we obtain:

$$\mathcal{L}\Big[\Phi(\tau)\Big]'(s) = \bar{\lambda}\mathcal{L}\Big[\mu(\tau)\Big]'(s) - \frac{2\alpha\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2})}{k_2^2(s+\frac{1}{k_2})^3} + \frac{\alpha\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})}{k_2^2(s+\frac{1}{k_2})^2} + \mathcal{L}\Big[\mu(\tau)\Big]'(s)\mathcal{L}\Big[\Phi(\tau)\Big](s) + \mathcal{L}\Big[\mu(\tau)\Big](s)\mathcal{L}\Big[\Phi(\tau)\Big]'(s). \tag{2.52}$$

By substituting $s = \frac{1}{k_2}$ in (2.52), rearranging and using (2.43) for $\mathcal{L}\Big[\mu(\tau)\Big](s)$ and $\mathcal{L}\Big[\mu(\tau)\Big]'(s)$, we obtain:

$$\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2}) + \frac{\alpha k_2}{(2-\alpha)}\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) = \frac{-\bar{\lambda}k_2\alpha}{2(2-\alpha)}. \tag{2.53}$$

(2.51) and (2.53) form a system of two equations with two variables, $\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2})$ and $\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})$. Solving this system leads to:

$$\mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) = \frac{\bar{\lambda}\alpha}{4(1-\alpha)},$$
$$\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2}) = \frac{-\bar{\lambda}\alpha k_2}{4(1-\alpha)}.$$

We can now solve (2.50). After rearranging:

$$\begin{aligned}\mathcal{L}\Big[\Phi(\tau)\Big](s) &= \frac{\alpha[\bar{\lambda} + \mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) + \frac{\sqrt{\alpha}}{k_2}\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})]}{k_2^2(s+\frac{1+\sqrt{\alpha}}{k_2})(s+\frac{1-\sqrt{\alpha}}{k_2})} - \frac{\alpha\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})}{k_2^2(s+\frac{1-\sqrt{\alpha}}{k_2})}\\ &= \frac{-\sqrt{\alpha}[\bar{\lambda} + \mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) + \frac{\sqrt{\alpha}}{k_2}\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})]}{2k_2(s+\frac{1+\sqrt{\alpha}}{k_2})}\\ &+ \frac{\sqrt{\alpha}[\bar{\lambda} + \mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) - \frac{\sqrt{\alpha}}{k_2}\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})]}{2k_2(s+\frac{1-\sqrt{\alpha}}{k_2})}.\end{aligned}$$

3. Inverse Laplace transform of $\Phi(\tau)$

We turn the Laplace transform of $\Phi(\tau)$ into the temporal domain.
For $k_1 = 2$, we can show that:

$$\Phi(\tau) = D_1\exp(-\omega_1\tau) + D_2\exp(-\omega_2\tau), \tag{2.54}$$

where:

$$\omega_1 = \frac{1+\sqrt{\alpha}}{k_2}, \tag{2.55}$$

$$\omega_2 = \frac{1-\sqrt{\alpha}}{k_2}, \tag{2.56}$$

$$\begin{aligned} D_1 &= \frac{-\sqrt{\alpha}[\bar{\lambda} + \mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) + \frac{\sqrt{\alpha}}{k_2}\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})]}{2k_2} \\ &= \frac{-\sqrt{\alpha}\bar{\lambda}[4-3\alpha-\sqrt{\alpha}\alpha]}{8k_2(1-\alpha)^2}, \end{aligned} \tag{2.57}$$

$$\begin{aligned} D_2 &= \frac{\sqrt{\alpha}[\bar{\lambda} + \mathcal{L}\Big[\Phi(\tau)\Big](\frac{1}{k_2}) - \frac{\sqrt{\alpha}}{k_2}\mathcal{L}\Big[\Phi(\tau)\Big]'(\frac{1}{k_2})]}{2k_2} \\ &= \frac{\sqrt{\alpha}\bar{\lambda}[4-3\alpha+\sqrt{\alpha}\alpha]}{8k_2(1-\alpha)^2}. \end{aligned} \tag{2.58}$$

4. Calculation of $\mathbb{V}(\widetilde{N}(t))$ from $\Phi(\tau)$

   From Equations (2.48) and (2.54), we calculate $\mathbb{V}(\widetilde{N}(t))$.

□

Chapter 3

Parameters of the recommendation system and performance indicators

3.1 Objective

In Chapter 1, we provided a baseline of the probability for a customer to accept a recommendation, based on their background information, meant for being used as the background intensity of the point process modelling the event “Subscription to a new insurance cover”. A method to make this baseline explainable, i.e. the SHAP values, was also developed. In Chapter 2, we developed a framework for MHP, including the definition, a method to estimate parameters and several mathematical properties.

This Chapter 3 makes the connection between the two previous chapters, by introducing the final MHP which models the recommendation system. It presents a complete parametrization of the MHP, by explaining the selection of life events integrated into the point processes, the building of background intensities or the values of hyperparameters. Then we propose an interpretation of the parameters obtained from the estimation detailed in Chapter 2. Finally, several performance indicators of the recommendation system are presented, in order to demonstrate the different contributions of the modelling by a MHP.

Chapter 3 is organized as follows. Section 3.2 presents the parametrization of the MHP. Section 3.3 details the results of the parameters estimation and proposes an interpretation of these results. Section 3.4 presents several performance indicators of the recommendation system.

3.2 MHP parametrization

3.2.1 List of life events

The first parameter to select is the list of life events to consider. The life events must have an impact on the probability to add an insurance cover, even indirectly. Since each event could influence another in a MHP, we could consider an event A which impacts the occurrence of an event B which makes the new subscription more or less likely.

Moreover, the life events must have a satisfying data quality. They must be observed from the beginning of the common observation period and on every customer considered. The method of data collection and the source of data should be reliable, whether they are owned by Foyer or open data.

Finally, the dimensions of the problem should be limited so that it does not affect the performances of the parameters estimation. Complexity increases quadratically with the number of events to consider m. To deal with high-dimensional problems, the authors in [86] propose a low-rank approximation of the kernel matrix to perform a nonparametric learning of the MHP.

Considering these three criteria, we selected the following list of life events, detailed in Definition 12.

Definition 12 ***(List of life events).***
We consider $m = 6$ types of events. Each number of event will be used for their corresponding intensity, or any mathematical object referring to a specific event. For instance, $\lambda_1^*(\cdot)$ represents the intensity of the event number 1.

- ***Event number 1:*** *Subscription to a new insurance cover, for which the intensity would represent the probability for the customer to accept a recommendation;*
- ***Event number 2:*** *Change of vehicle;*
- ***Event number 3:*** *Modification of household composition (including weddings);*
- ***Event number 4:*** *Births;*
- ***Event number 5:*** *Change of job;*
- ***Event number 6:*** *Move.*

3.2.2 Background intensity

In Definition 2, we introduced the background intensity $\lambda_i(\boldsymbol{x_u}(t))$, for each event $i \in \{1, ..., m\}$ and customer $u \in \mathcal{U}$. In Section 2.3.1, we assumed that $\lambda_i(\boldsymbol{x_u}(t))$ is decomposed into the product of a common proportional factor μ_i and $f_i(\boldsymbol{x_u}(t))$ which introduces a personalization for each customer (see Equation (2.8)). We present below the building of each $f_i(\boldsymbol{x_u}(t))$:

- **Event number 1:** $f_1(\boldsymbol{x_u}(t))$ has been estimated in Chapter 1. For every customer $u \in \mathcal{U}$, $f_1(\boldsymbol{x_u}(t)) = \hat{f}(x_u(t))$, where $\hat{f}(x_u(t))$ is introduced in Section 1.4;

- **Event number 2:** $f_2(\boldsymbol{x_u}(t))$ is estimated by an Extreme Gradient Boosting algorithm, which learned the vehicle change frequency over customers who subscribed to a car insurance for five years at least. The features taken into account are mainly the vehicle characteristics and the customer's profile;

- **Event number 3:** $f_3(\boldsymbol{x_u}(t)) = 1$ for all customer $u \in \mathcal{U}$. A modification of household composition includes many types of events (e.g. flatsharing, civil union, weddings, step-families), implies complex interactions and could occur at any moment of life. That's why we assume that background intensity is equal for each customer;

- **Event number 4:** $f_4(\boldsymbol{x_u}(t))$ equals the birthrate of the age group the customer u belongs to. Birthrates were extracted from the national statistics portal of Luxembourg [87]. In Appendix B, Table B.1 presents fertility rates by age groups in Luxembourg;

- **Event number 5:** $f_5(\boldsymbol{x_u}(t))$ is built from a survey of the French Institute of Statistics (see [88]), assuming that France and Luxembourg have similar population dynamics as France is a border country. $f_5(\boldsymbol{x_u}(t))$ equals the rate of people who changed of job over five years by age group. In Appendix C, Table C.1 presents the change of job rates by age groups in France observed from 2010 to 2015;

- **Event number 6:** $f_6(\boldsymbol{x_u}(t)) = 1$ for all customer $u \in \mathcal{U}$. Several studies indeed show that moving out is almost always linked to a life event: 70% are due to family events (births, weddings...) and 30% to professional reasons according to the French Institute of Statistics (see [89]). Thus, we assume that the background intensity is the same for every customer and the event is only influenced by the occurrence of life events.

3.2.3 Data and hyperparameters

This work uses data provided by Foyer Assurances. We extracted sequences of life events from Foyer customers observed between $T_1 = 2015$ and $T_2 = 2018$ included. Each row from the final data-set represents one event. The columns are:

- Customer's ID number $u \in \mathcal{U}$;
- Time length we observe life events for customer u, T_u;
- Type of event (e_k^u denotes the k^{th} event that occurred to customer u);
- Time of occurrence (t_k^u denotes the time when the k^{th} event occurred to customer u).

Some descriptive statistics about this data-set are listed below:

- There are around $U = 63.000$ customers;
- The number of rows is $\sum_{u=1}^{U} N^u(T_2) = 149.000$ approximately, therefore there are around 2.4 events per customer on average;
- We also present the distributions of events in Table 3.1 below.

For hyperparameters introduced in the previous section we set the following values:

- $N_{iter} = 100$, $\varepsilon = 0.005$, values used in [80];
- $\omega = 0.05$, fixed from the rule of thumb detailed in [80];
- $\kappa = 2$, as developed in [2] which allows to include a delay between the occurrence of an event and its effect on the other types of events;
- $D = 6$, so that immediate response and five different delayed triggers are taken into account. As we can see in Figure 3.1, functions curves flatten when d increases, therefore adding more basis functions would not provide a type of response distinct from the first six functions;
- $\alpha_1 = 10^4$;
- $\alpha_2 = 10^6$.

To set α_1 and α_2, we performed a grid-search over a parameter grid based on NLL value, where $(\alpha_1, \alpha_2) \in [10^0, 10^8]^2$.

In Figure 3.1 we plot the D basis functions $g_d, d \in \{1, ..., D\}$.

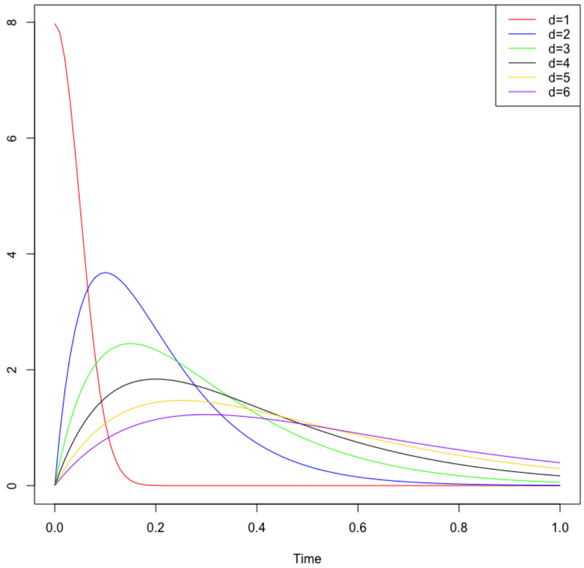


Figure 3.1: Plot of basis functions $g_d, d \in \{1, ..., D\}$

3.3 Parameters estimation

3.3.1 Results

The parameters estimation leads to the following results:

- $\hat{\boldsymbol{\mu}} = \left(\hat{\mu}_i\right)_{i \in \{1,...,m\}} = [2.74 \times 10^{-2}, 6.76 \times 10^{-1}, 8.65 \times 10^{-2}, 2.11 \times 10^{-4}, 1.52 \times 10^{-3}, 8.24 \times 10^{-2}]$, as described in Equation (2.8) where the background intensity is proportional to $\hat{\boldsymbol{\mu}}$;
- $\hat{\boldsymbol{A}}_{\boldsymbol{d}} = \left(\hat{a}^d_{i,j}\right)_{i,j=1,...,m}$, $\hat{\boldsymbol{A}} = \{\hat{\boldsymbol{A}}_{\boldsymbol{d}}\}_{d \in \{1,...,D\}}$: in Table 3.1 we compute $\hat{\boldsymbol{A}}^{sum}$, estimation of $\boldsymbol{A}^{sum} = \sum_{d=1}^{D} \boldsymbol{A}_{\boldsymbol{d}} = \left(a^{sum}_{i,j}\right)_{i,j=1,...,m}$, the sum of the D infectivity matrix. As a reminder, each $a^d_{i,j}$ is the coefficient used to calculate the d^{th} triggering function for the influence of event i on event j (see Equation (2.9)).

Triggered \ **Triggering**	Ins. cov. modif.	Change of vehicle	House. compo.	Birth	Change of job	Move
Insurance cover modif.	1.4×10^{-2}	4.4×10^{-2}	8.2×10^{-4}	3.4×10^{-6}	2.2×10^{-3}	1.65×10^{-3}
Change of vehicle	1.7×10^{-15}	6.3×10^{-15}	1.5×10^{-5}	4.8×10^{-4}	1.1×10^{-3}	7.3×10^{-8}
Household composition	2.5×10^{-3}	8.6×10^{-3}	6.2×10^{-23}	5.4×10^{-3}	1.0×10^{-2}	3.0×10^{-2}
Birth	9.6×10^{-5}	6.8×10^{-5}	3.1×10^{-5}	3.9×10^{-15}	1.5×10^{-3}	3.6×10^{-3}
Change of job	1.0×10^{-3}	2.0×10^{-4}	5.6×10^{-4}	9.7×10^{-4}	1.2×10^{-2}	2.0×10^{-3}
Move	9.7×10^{-4}	6.5×10^{-7}	4.2×10^{-3}	4.9×10^{-4}	2.8×10^{-3}	1.8×10^{-18}

Table 3.1: Estimation of the sum of the D infectivity matrix $\boldsymbol{A}^{sum}$

All the matrix $\hat{\boldsymbol{A}}_{\boldsymbol{d}}$ are in Appendix D. We propose explanations and an interpretation of these numerical results in the following subsection.

3.3.2 Interpretation

We propose below an interpretation of the results of parameters estimation.

3.3.2.1 Highest interactions between events

The matrix $\boldsymbol{A}^{sum}$ models the way events trigger each other. The higher is $a_{i,j}^{sum}$, the higher is the increasing of the intensity of event i when j occurred. We sort in Table 3.2 the interactions Event $i \rightarrow$ Event j by decreasing coefficients $\hat{a}_{i,j}^{sum}$.

Event triggering	**Event triggered**	**Coefficient**
Change of vehicle	Insurance cover modification	4.45×10^{-2}
Move	Household composition	3.07×10^{-2}
Insurance cover modification	Insurance cover modification	1.46×10^{-2}
Change of job	Change of job	1.29×10^{-2}
Change of job	Household composition	1.02×10^{-2}
Change of vehicle	Household composition	8.64×10^{-3}
Birth	Household composition	5.49×10^{-3}
Household composition	Move	4.23×10^{-3}
Move	Birth	3.61×10^{-3}
Change of job	Move	2.86×10^{-3}

Table 3.2: Highest interactions between events

We explicit the meaning of each coefficient $\hat{a}_{i,j}^{sum}$. The first coefficient of Table 3.2, i.e. $\hat{a}_{1,2}^{sum}$, represents how the intensity of the event "Insurance cover modification" is influenced when the event "Change of vehicle" occurs. As described in Equation (2.9), the coefficient $\hat{a}_{1,2}^{sum}$ is the sum of $\hat{a}_{i,j}^{d}, d \in \{1, ..., D\}$, weighting the basis triggering functions. The fact that it is the highest coefficient means that among all types of events, the most significant jump of intensity when an event occurs is when a customer changes his vehicle, which increases his probability to modify his insurance cover.

We notice that insurance cover modifications are likely to be triggered by a change of vehicle and by a previous modification. It means that according to our model, we should recommend a new guarantee when the customer buys a new car (a different vehicle implies new needs of insurance) or has recently modified his cover (the customer is more receptive after a first change). Other high interactions also make sense, such as the link between move and household composition: people often get a common place when they move in together.

3.3.2.2 Expected value

For each customer u and type of event i, assuming N_i^u is a stationary process, the expected total of events at the end of observation period is denoted n_i^u. We also denote $\boldsymbol{n^u} = \left[n_i^u\right]_{i\in\{1,\dots,m\}}$. $\boldsymbol{n^u}$ is given by (see [90]):

$$\boldsymbol{n^u} = T_u\Big(I_m - \boldsymbol{A}^{sum}\Big)^{-1}\boldsymbol{\lambda(x_u)}, \tag{3.1}$$

where I_m is the identity matrix of dimensions $m \times m$. This first order moment allows us to check whether the order of magnitude of parameters is correct, by comparing the theoretical expected value of the total of events and the distribution observed in the data-set. We could also quantify the average part of occurrences triggered by past events, for event i and customer u, by computing $\frac{n_i^u - \lambda_i(\boldsymbol{x_u})T_u}{n_i^u}$. On the opposite, the quotient $\frac{\lambda_i(\boldsymbol{x_u})T_u}{n_i^u}$ represents the part of events which occur due to the "natural" tendency of the customer, modelled by the background intensity. In Table 3.3, we present:

- The distribution of events observed in the data-set, on average per customer: $\frac{1}{U}\sum_{u=1}^{U} N_i^u(T_u)$;
- The average of expected value of the number of events, over every customer: $\frac{1}{U}\sum_{u=1}^{U} n_i^u$;
- The average of background intensities multiplied by time exposure, over every customer: $\frac{1}{U}\sum_{u=1}^{U} \lambda_i(\boldsymbol{x_u})T_u$;
- The average part of occurrences triggered by past events, over every customer: $\frac{1}{U}\sum_{u=1}^{U} \frac{n_i^u - \lambda_i(\boldsymbol{x_u})T_u}{n_i^u}$.

Type of event	**Observed**	**Expected value**	**Background intensity**	**% from past events**
Insurance cover modif.	9.70×10^{-2}	9.16×10^{-2}	1.50×10^{-2}	83.6%
Change of vehicle	1.22	1.21	1.21	0.01%
Household compo.	4.18×10^{-1}	4.35×10^{-1}	4.11×10^{-1}	5.66%
Birth	5.64×10^{-2}	5.36×10^{-2}	5.18×10^{-2}	3.29%
Change of job	1.63×10^{-1}	1.52×10^{-1}	1.49×10^{-1}	2.22%
Move	4.32×10^{-1}	3.93×10^{-1}	3.91×10^{-1}	0.60%

Table 3.3: Comparison between the distribution of events observed in the data-set and the expected total of events

First, we observe that the order of magnitude of parameters is correct, by comparing actual average of events with their expected value. Estimation on event number 1 seems more accurate than other events, which could be explained by the fact that change of vehicle represents half of total events. We also notice that according to our model, 83% of insurance cover modifications are triggered by other life events. This observation justifies our model a posteriori, since the objective is to take into account the influence of life events on this type of event. We could also observe that on the opposite the changes of vehicle are almost never triggered by other events. This seems to be in compliance with Luxembourgish customers behavior: they buy a new vehicle every 3-4 years without waiting for a specific event.

3.3.2.3 Prediction by simulation

The main interest of building our recommendation system with a Multivariate Hawkes Process is that the recommendations are updated with respect to events occurrences, which allows us to have the best timing to suggest insurance covers to customers. The recommendation system is not based on expected values provided by the model, but on instantaneous intensities updated by events occurrences. However, it is useful to know the distribution of the total of events for each customer for other purposes than marketing (for instance, forecasting the workload of company's staff, which is proportional to the total of events to manage). We evaluate the distribution numerically by simulating n trajectories and calculating empirical distribution.

In order to simulate the estimated Multivariate Hawkes Process for customer u from time 0 to T_u, we use the Ogata's thinning algorithm (see [77]), written in Algorithm 4 in Appendix.

We perform simulations on two types of customers, denoted u_1 and u_2:

- a younger customer aged 29, more likely to change his vehicle, to have a child and get a new job spontaneously (i.e. according to the background intensities);
- an older customer aged 45, less likely to change his vehicle, to have a child and get a new job spontaneously.

Moreover, both have a similar background intensity for the event "Subscription to a new insurance cover", i.e. they have the same probability to modify their cover regardless of life events. We plot in Figures 3.2 and 3.3 the simulated distribution of events over 10 000 simulations for both customers with the following sets of parameters: $T_{u_1} = 5, f_1(\boldsymbol{x_{u_1}}) = 0.12, f_2(\boldsymbol{x_{u_1}}) = 0.42, f_4(\boldsymbol{x_{u_1}}) = 77.6, f_5(\boldsymbol{x_{u_1}}) = 33$ and $T_{u_2} = 5, f_1(\boldsymbol{x_{u_2}}) = 0.12, f_2(\boldsymbol{x_{u_2}}) = 0.32, f_4(\boldsymbol{x_{u_2}}) = 47.97, f_5(\boldsymbol{x_{u_2}}) = 16$, estimated as detailed in Section 3.2.2.

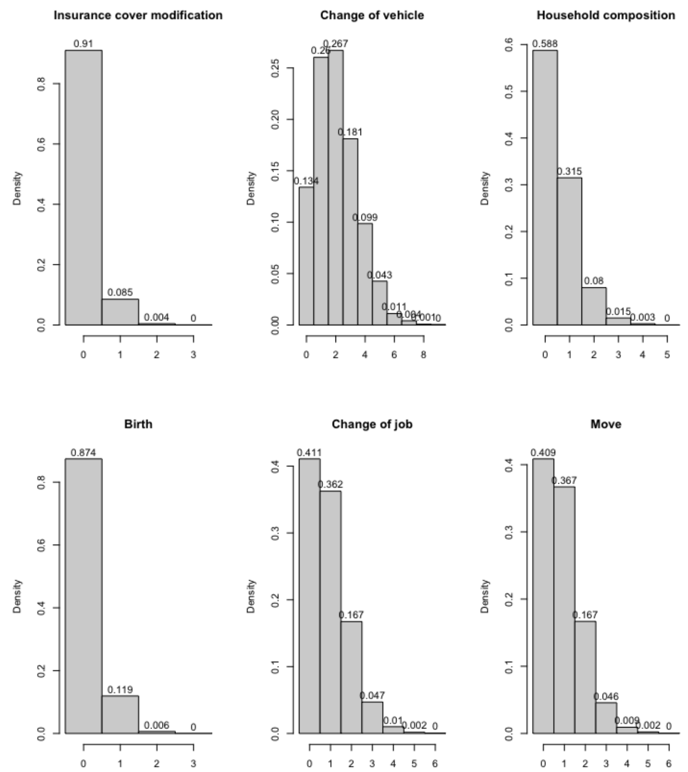


Figure 3.2: Distribution of events over 10,000 simulations for a 29-year-old customer with parameters: $T_{u_1} = 5, f_1(\boldsymbol{x_{u_1}}) = 0.12, f_2(\boldsymbol{x_{u_1}}) = 0.42, f_4(\boldsymbol{x_{u_1}}) = 77.6, f_5(\boldsymbol{x_{u_1}}) = 33$.

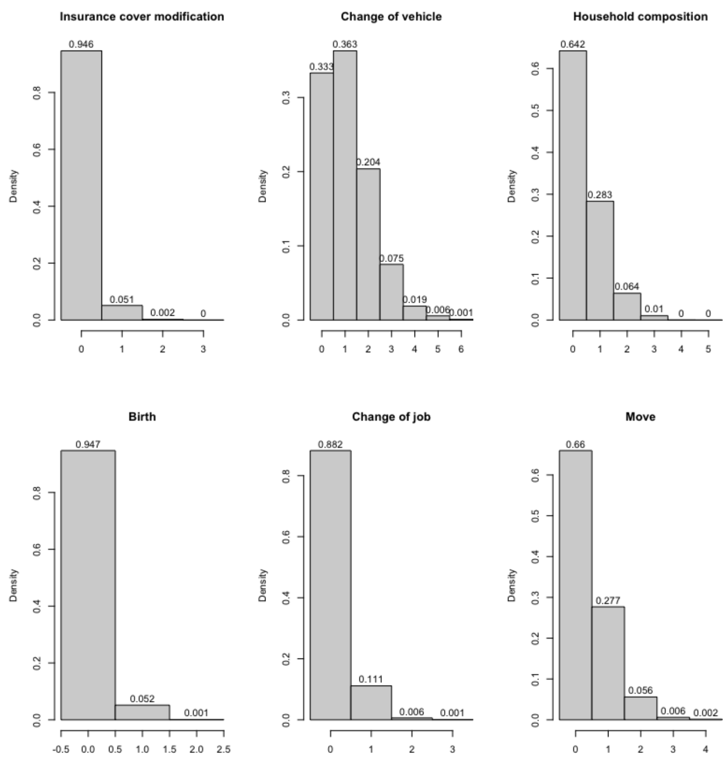


Figure 3.3: Distribution of events over 10,000 simulations for a 45-year-old customer with parameters: $T_{u_2} = 5, f_1(\boldsymbol{x_{u_2}}) = 0.12, f_2(\boldsymbol{x_{u_2}}) = 0.32, f_4(\boldsymbol{x_{u_2}}) = 47.97, f_5(\boldsymbol{x_{u_2}}) = 16$.

We compare results of simulations for customers u_1 and u_2. We notice that despite even background intensities, the customers u_1 and u_2 don't have the same probability to add a guarantee. The probability that the customer u_1 adds a guarantee is 70% higher than for the customer u_2 (i.e. 9.1% versus 5.3%). This illustrates the influence of life events on the subscription to an insurance cover: with an even background intensity, the difference between the two customers is significant. It could be explained by the higher frequency of events observed for the customer u_1.

Regarding each customer apart, the customer u_1 has a 58.9% probability to change of job over the next five years and a 59.1% chance to move in another place for instance, which seems correct as order of magnitude. Concerning the customer u_2, events distributions make sense as well given customer's age and situation: this customer aged 45 and married is not likely to become a father again (5.3%), to have a new job (11.8%) and he is more likely to keep his household steady (66% of probability to keep the same household composition and 64% of probability to keep the same apartment/house).

3.3.2.4 Influence of regularization

As a reminder, regularization aims to correct the main default observed on the data-set, which is the inaccuracy of times of occurrence. Some events might happen before some others according to the data-set, while it is the opposite in real life. The consequence of this inaccuracy may be that these misreported dates could reinforce the wrong coefficients from triggering matrix. The objective of this subsection is to show that regularization allows us to fix this problem.

To do so, we compare the list of higher interactions without and with regularization. Especially, we check whether couples of events (Event 1, Event 2) which appear on top of the list with both causations Event 1 $\rightarrow$ Event 2 and Event 2 $\rightarrow$ Event 1 without regularization appear with only one causation with regularization. Table 3.4 compares higher interactions (as defined in Section 3.3) in both cases by highlighting the various couples of events.

Without regularization			**With regularization**		
Triggering	**Triggered**	**Coefficient**	**Triggering**	**Triggered**	**Coefficient**
Move	Household composition	5.90×10^{-2}	Change of vehicle	Insurance cover modification	4.45×10^{-2}
Change of vehicle	Insurance cover modification	5.38×10^{-2}	Move	Household composition	3.07×10^{-2}
Insurance cover modification	Change of vehicle	4.89×10^{-2}	Insurance cover modification	Insurance cover modification	1.46×10^{-2}
Insurance cover modification	Insurance cover modification	4.70×10^{-2}	Change of job	Change of job	1.29×10^{-2}
Change of vehicle	Household composition	2.39×10^{-2}	Change of job	Household composition	1.02×10^{-2}
Change of job	Change of job	2.35×10^{-2}	Change of vehicle	Household composition	8.64×10^{-3}
Household composition	Move	1.91×10^{-2}	Birth	Household composition	5.49×10^{-3}
Move	Birth	1.82×10^{-2}	Household composition	Move	4.23×10^{-3}
Change of job	Household composition	1.35×10^{-2}	Move	Birth	3.61×10^{-3}
Move	Insurance cover modification	8.40×10^{-3}	Change of job	Move	2.86×10^{-3}

Table 3.4: Comparison of highest interactions between events without and with regularization

We notice that with regularization, only one causation from couples of events (Move, Household composition) and (Change of vehicle, Insurance cover modification) is kept as expected. The other causation has a lower coefficient, which allows us to define which event triggers the other for every couple.

3.4 Backtesting over year 2019

In order to check the accuracy of our recommendation system, we perform a backtesting over year 2019 as follows:

1. We consider the Multivariate Hawkes Process learned as described previously from data observed from 2015 to 2018, whose parameters are given in Section 3.3. For each customer u, we focus on $\lambda_1^{u,*}(t)$, the intensity related to the event 1 (i.e. subscription to a new insurance cover), where t takes its values on year 2019;

2. Over 2019, we observe $\lambda_1^{u,*}(t)$, considering that every event of types 2 to 6 in historical data occurs on every customer u. We calculate recommendations by two methods:

   - We consider that any time the intensity is higher than a threshold $\lambda^*_{\text{threshold}}$, we should recommend to the customer the upgrade of his insurance cover. This threshold equals the quantile of the maximal intensity observed on each customer in 2019 of order $1-\alpha$, where α is the proportion of customers who changed their cover over year 2019:

   $$\lambda^*_{\text{threshold}} = m(\{\max_t \left(\lambda_1^{u,*}(t)\right)\}_{u\in\mathcal{U}}, \lfloor \alpha U \rfloor), \tag{3.2}$$

   where $m(S,n)$ is the subset of S which contains elements of S higher that the n^{th} maximum of S. Choosing this threshold allows us to get the same amount of recommendations than observed in 2019;

   - We consider that we should make recommendations only based on background intensities, i.e. on customers u with $\lambda_1(\boldsymbol{x_u})$ higher than a threshold $\lambda_{\text{threshold}}$ (in a similar way than in [1]) fixed so that the two methods propose the same total of recommendations:

   $$\lambda_{\text{threshold}} = m(\{\lambda_1(\boldsymbol{x_u})\}_{u\in\mathcal{U}}, \lfloor \alpha U \rfloor); \tag{3.3}$$

3. To quantify the performances of the recommendations, we compute the confusion matrix which allows us to compare actual and predicted insurance cover subscriptions. We represent the confusion matrix for both methods respectively in Tables 3.5 and 3.6.

Actual/Predicted	**No subscription**	**Subscription**
No subscription	56 734 (82%)	3401 (4%)
Subscription	3401 (4%)	5548 (8%)

Table 3.5: Confusion matrix to compare actual and predicted insurance cover subscriptions over year 2019 based on full intensity

Actual/Predicted	**No subscription**	**Subscription**
No subscription	56 734 (82%)	3401 (4%)
Subscription	5907 (8%)	3042 (4%)

Table 3.6: Confusion matrix to compare actual and predicted insurance cover subscriptions over year 2019 based on background intensity

We see that our Multivariate Hawkes Process allows us to compute accurate recommendations. The confusion matrix shows a sensitivity of 62% with the first method (i.e. the proportion of recommendations that are correctly identified). Moreover, the comparison with the second method shows that the influence of life events allows us to improve significantly the recommendation system accuracy. The accuracy drops to 34% without considering the impact of life events.

We focus on customers on which the MHP approach allows us to turn a wrong recommendation into a right one, in order to understand for which category of customers the MHP performs better than the background intensity alone. In Table 3.7, we compare the average number of events per customer for the whole portfolio of customers and for customers correctly detected by the full intensity of the MHP for recommendation but not by the background intensity (i.e. those among the 5548 customers correctly recommended in Table 3.5 but not among the 3042 customers identified in Table 3.6).

Type of event	**All customers**	**Customers detected by the MHP**
Change of vehicle	1.22	3.03
Household composition	4.18×10^{-1}	5.64×10^{-1}
Birth	5.64×10^{-2}	6.35×10^{-2}
Change of job	1.63×10^{-1}	2.22×10^{-1}
Move	4.32×10^{-1}	4.31×10^{-1}

Table 3.7: Comparison of the average number of events of customers correctly detected by the full intensity of the MHP for recommendation but not by the background intensity

For instance, among the customers correctly detected by the full intensity but not by the background intensity, the average number of changes of vehicle is around 3, while on the entire portfolio this average number drops to 1.2. We see that the MHP performs well on customers who change their vehicle more frequently than the others. As observed in Table 3.2 listing the highest interactions between events, a change of vehicle leads to a higher intensity for the process “Subscription to a new insurance cover”, which allows the recommendation system to detect the new guarantees linked to a new vehicle.

Chapter 4

R Package

4.1 Objective

The objective of Chapter 4 is to introduce `[libraryname]`, a R package that we developed in order to provide to the R community to build a recommendation system based on a MHP and a wide range of plotting tools around MHP. It proposes an implementation of the parameters estimation algorithm presented in Section 2.4.2, of the Ogata's thinning algorithm (see Section 2.2.5.2) and a computation of MHP moments (see Section 3.3.2.2). Several plot functions are included, in order to clearly display as much information as possible about MHP. The `[libraryname]` package provides also examples and a full documentation.

4.2 Existing libraries

Despite the fact that Hawkes processes are studied in numerous papers in the recent literature, there exists no R packages which implement functions to estimate, plot or simulate MHP by letting the choice of the triggering functions in parameters. The package `hawkes` [29] computes likelihood, expectation and variance of the process, and proposes an implementation of Ogata's simulation algorithm. The package `emhawkes` [30] proposes an Expectation-Maximization algorithm implementation via several maximization methods such as Newton-Raphson, Broyden-Fletcher-Goldfarb-Shanno, Berndt-Hall-Hall-Hausman, Simulated ANNealing, Conjugate Gradients or Nelder-Mead (see [91]). However these two packages include no plotting methods and they are restricted to exponential triggering functions. The package `hawkesbow` [31] includes the possibility to choose different functions (e.g. power law function, Gaussian or Pareto density) and a plotting function representing the intensity and the dates of occurrence. However only the univariate case is implemented. Also, the package `PtProcess` [32] is designed to model earthquakes by marked point processes and includes only univariate analysis as a consequence.

Regarding Python libraries about MHP, pyhawkes [33] includes Bayesian inference methods to estimate a Hawkes process in discrete time. The library `tick` [34] proposes both parametric and non-parametric Hawkes process estimation for the multivariate case and several plot tools, in order to represent the intensity, the triggering functions or the events history.

4.3 Features

4.3.1 Package architecture

The package `[libraryname]` includes several types of functions, which implement most of the elements studied in Chapters 1, 2 and 3. Here is a short description of the different categories of functions:

- Data preparation: loading and preparing data from raw datasets, so that all the content is in the right format for further analysis;
- Parameters estimation: implementing the parameters estimation described in Section 2.4;
- Simulation: simulating a MHP according to Algorithm 4;
- Mathematical objects: computing quantities such as expectations, variances, functions (e.g. intensity, triggering functions);
- Plotting: displaying results quickly such as intensity, events history, SHAP values, etc.

Table 4.1 details the full structure of the package `[libraryname]` by proposing an exhaustive list of the functions and their description.

Category	Function	Description
Data preparation	`load-data`	Import a raw dataset and turn it into a list where each element contains the sequence of past events and information about a customer
	`initialization`	Create a model with parameters selected randomly from loaded data in order to initialize model learning
Parameters estimation	`learn-MLE`	Learn model parameters according to the algorithm described in Section 2.4 in function of several assumptions (e.g. penalization, triggering functions)
Simulation	`simulation`	Simulate a MHP according to Algorithm 4
Mathematical objects	`intensity`	Calculate the intensity function of a MHP given events history
	`MHP-expectation`	Calculate the expectation of MHP based on parameters, according to Equation (3.1)
Plotting	`plot-history`	Plot the history of both intensity and events occurrences of any process from a MHP

Table 4.1: Description of the full structure of the package `[libraryname]`

4.3.2 Estimation and simulation: comparison with R packages

To our knowledge, the package `[libraryname]` is the first R package which implements a parameters estimation for MHP with several observed parses. Data used to build the recommendation system are divided into sequences, where one sequence corresponds to one customer. In Section 2.4, estimation algorithm is based on all the sequences of customers $u \in \mathcal{U}$. The R packages mentioned in Section 4.2, which propose an implementation of estimation for MHP, learn parameters only from one sequence of events (e.g. the occurrence times of earthquakes in a given area). As a consequence, we cannot compare the performances of `[libraryname]` with another package in terms of computational time.

However, we could compare the performances between the simulation algorithms from the packages by adapting the parameters of the MHP. Since the package `hawkes` allows an exponential triggering function only, we consider a MHP related to a customer u, which includes m processes and is defined by, for all $i \in \{1, ..., m\}$:

$$\begin{aligned}
\lambda_i^{u,*}(t) &= \lambda_i(\boldsymbol{x}_{\boldsymbol{u}}(t)) + \sum_{j=1}^{m} \sum_{k=1}^{N_j^u(t)} \mu_{i,j}(t - t_j^k), && (4.1)\\
\mu_{i,j}(t) &= \alpha_{i,j} \frac{\exp(-\frac{t}{\omega})}{\omega}, && (4.2)\\
\alpha_{i,j} &= a_{i,j}^1. && (4.3)
\end{aligned}$$

We select $u = u_1$ (i.e. the 29-year-old customer simulated in Figure3.3) and we simulate five years of life events by using the following packages: `[libraryname]`, `hawkes`.

```
# Loading the packages and the model
library([libraryname])
library(hawkes)
load(model)

# Parameters
n=10000 #number of simulations
horizon=5 #projection over 5 years
mu=model$mu
A=model$A
w=model$w

# Simulations
t_1=Sys.time()
simulation(model,horizon,n) #package [libraryname]
t_2=Sys.time()
time_1=t_2-t_1

t_1=Sys.time()
for(i in 1:n)
    simulateHawkes(mu,A,w,horizon) #package hawkes
t_2=Sys.time()
time_2=t_2-t_1
```

We compare the computational times in Figure 4.1, on logarithmic scale.

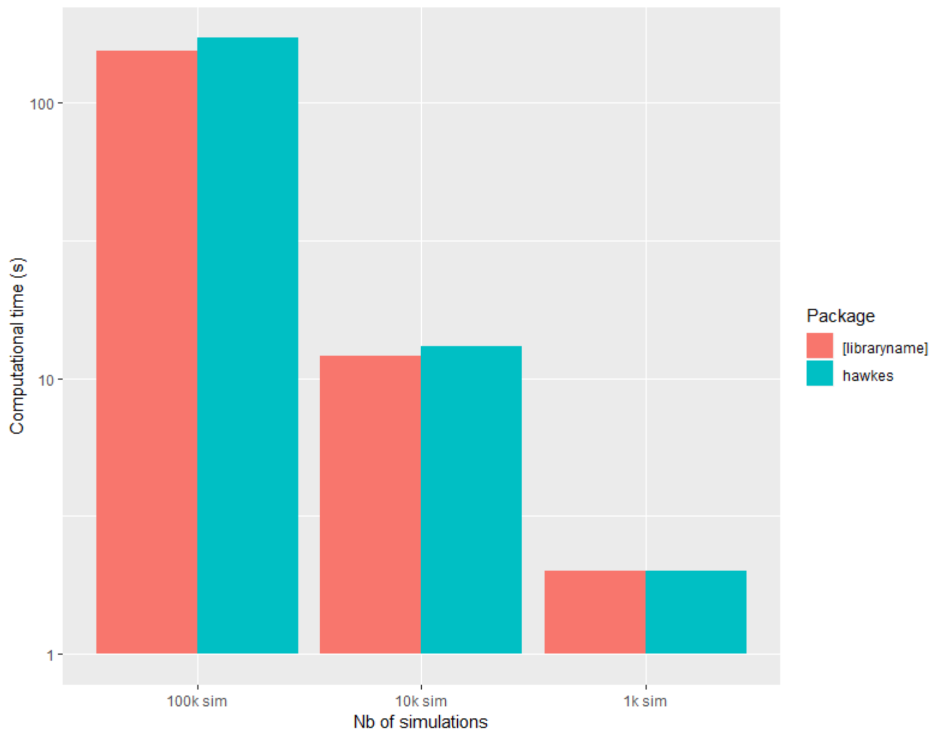


Figure 4.1: Comparison of the computational times for several numbers of simulations representing a 29-year-old customer with parameters: $T_{u_1} = 5, f_1(\boldsymbol{x}_{\boldsymbol{u_1}}) = 0.12, f_2(\boldsymbol{x}_{\boldsymbol{u_1}}) = 0.42, f_4(\boldsymbol{x}_{\boldsymbol{u_1}}) = 77.6, f_5(\boldsymbol{x}_{\boldsymbol{u_1}}) = 33$.

We could see that regarding simulations, the performances of the packages are equivalent, while optimizing simulations is not the prime objective of the package `[libraryname]`.

Chapter 5

Contributions and perspectives

5.1 Objectives

This chapter has two main objectives. The first aim is to sum up the contributions of the previous chapters, by presenting a synthesized view of the novelties introduced in our work. The second aim is to conclude on the perspectives considered for future work.

It is organized as follows. Section 5.2 lists the main contributions of the thesis, where each novelty is described into a corresponding subsection. Section 5.3 presents several leads for future work about the recommendation system based on a MHP.

5.2 List of contributions

5.2.1 Recommendation of insurance guarantees via the Apriori algorithm

Through Assumption 2, we considered that the two following questions could be separated:

- To whom should we recommend an additional cover?
- Which cover should be recommended?

The second question is answered thanks to Apriori, whose application to insurance is a novelty. Apriori method, synthesized in Algorithm 3, consists in generating all the association rules between an existing cover and all the unsubscribed guarantees and selecting the one with the highest confidence. The comparison with other algorithms of type collaborative filtering (i.e. users-items associations for recommendation systems) reveal that Apriori is the most accurate approach as seen in Table 1.4:

Method	Accuracy
Random	49 %
Popular	67 %
IBCF	71 %
UBCF	82 %
SVD	72 %
Apriori	95 %

5.2.2 Integration of life events for recommendation system

To the best of our knowledge, in the literature there exists no recommendation system based on personal life events. Building a recommendation system with real data about life events from customers, which leads to up-to-date recommendations in function of what happens to customers in day-to-day life, is a contribution of this thesis. The following life events are taken into account in the recommendation system (see Definition 12):

- Change of vehicle;
- Modification of household composition (including weddings);
- Births;
- Change of job;
- Move.

5.2.3 Personalized background intensity

For most of approaches presented in the related works (see Chapter 0), estimated background intensities of MHP are identical for every user. The recommendation system includes a personalized background intensity for each customer, based on either Machine Learning or relevant statistical analysis (see Section 3.2.2):

- Events "Subscription to a new insurance cover" and "Change of vehicle" have a background intensity estimated by a XGBoost algorithm, from a database including information about customers' profile. The algorithm trained for the first type of event was the first version of the recommendation system, tested on a pilot phase;
- Events "Birth" and "Change of job" have a background intensity estimated from statistic surveys, which make a distinction between age groups concerning the occurrence rate.

5.2.4 Suited triggering functions: Γ-Hawkes Processes properties

For the building of the MHP, we propose triggering functions which are adapted to the insurance data to our disposal and which are different from classic kernels. We first introduce the notion of Γ-Hawkes Processes, a category of univariate Hawkes Process with a triggering function of Gamma density. We demonstrate several mathematical properties about Γ-Hawkes Processes, whose conditional intensity function is:

$$\lambda^*(t) = \lambda + \sum_{k=1}^{N(t)} \mu(t - t_k),$$

and whose triggering function is $\mu(t) = \alpha \frac{t^{k_1-1}\exp(-\frac{t}{k_2})}{k_2^{k_1}\Gamma(k_1)}$, for several values of k_1:

- Expectation:

  1. For $k_1 = 1$ (i.e. $\mu(t) = \alpha \frac{\exp(-\frac{t}{k_2})}{k_2}$, exponential decay):

  $$\mathbb{E}(N(t)) = \frac{\lambda}{1-\alpha}t - \frac{\alpha\lambda k_2}{(1-\alpha)^2}\Big[1 - \exp\Big(-\frac{1-\alpha}{k_2}t\Big)\Big].$$

  2. For $k_1 = 2$ (i.e. $\mu(t) = \alpha \frac{t\exp(-\frac{t}{k_2})}{k_2^2}$):

  $$\begin{aligned}\mathbb{E}(N(t)) = \frac{\lambda}{1-\alpha}t &+ \frac{\lambda\sqrt{\alpha}k_2}{2(1+\sqrt{\alpha})^2}\Big[1 - \exp\Big(-\frac{1+\sqrt{\alpha}}{k_2}t\Big)\Big] \\ &- \frac{\lambda\sqrt{\alpha}k_2}{2(1-\sqrt{\alpha})^2}\Big[1 - \exp\Big(-\frac{1-\sqrt{\alpha}}{k_2}t\Big)\Big].\end{aligned}$$

  3. For $k_1 = 3$ (i.e. $\mu(t) = \alpha \frac{t^2\exp(-\frac{t}{k_2})}{k_2^3\Gamma(3)}$):

  $$\begin{aligned}\mathbb{E}(N(t)) = \frac{\lambda}{1-\alpha}t + \frac{\lambda\alpha B}{k_2^3}\exp(-r_1 t) &+ \frac{\lambda\alpha C}{k_2^3}\cos(\omega t)\exp(-at) \\ &+ \frac{\lambda\alpha(D - Ca)}{k_2^3\omega}\sin(\omega t)\exp(-at),\end{aligned}$$

  where

  $$B = \frac{1}{r_1(r_1a_1 - r_1^2 - a_2)}, \quad C = \frac{r_1 - a_1}{a_2(r_1a_1 - r_1^2 - a_2)}, \quad D = \frac{r_1a_1 - a_1^2 + a_2}{a_2(r_1a_1 - r_1^2 - a_2)},$$
  $$r_1 = \frac{1-\alpha^{1/3}}{k_2}, \quad a_1 = \frac{(2+\alpha^{1/3})}{k_2}, \quad a_2 = \frac{(1-\alpha)}{(1-\alpha^{1/3})k_2^2}, \quad a = \frac{(2+\alpha^{1/3})}{2k_2},$$
  $$\omega = \frac{\sqrt{3}\alpha^{1/3}}{2k_2}.$$

- Variance:

  1. For $k_1 = 1$ (i.e. $\mu(t) = \alpha\frac{\exp(-\frac{t}{k_2})}{k_2}$):

$$\mathbb{V}(N(t)) = \frac{\lambda}{(1-\alpha)^3}t - \frac{\lambda\alpha(2-\alpha)k_2}{(1-\alpha)^4}\Big[1 - \exp\Big(-\frac{1-\alpha}{k_2}t\Big)\Big].$$

  2. For $k_1 = 2$ (i.e. $\mu(t) = \alpha\frac{t\exp(-\frac{t}{k_2})}{k_2^2}$):

$$\begin{aligned}
\mathbb{V}(N(t)) &= C_1 + C_2 t + C_3\exp(-\omega_1 t) + C_4\exp(-\omega_2 t), \text{ where:}\\
\omega_1 &= \frac{1+\sqrt{\alpha}}{k_2},\\
\omega_2 &= \frac{1-\sqrt{\alpha}}{k_2},\\
C_1 &= \frac{-\alpha k_2\lambda[8-5\alpha+\alpha^2]}{2(1-\alpha)^4},\\
C_2 &= \frac{\lambda}{(1-\alpha)^3},\\
C_3 &= \frac{-\sqrt{\alpha}\lambda k_2[4-3\alpha-\sqrt{\alpha}\alpha]}{4(1-\alpha)^2(1+\sqrt{\alpha})^2},\\
C_4 &= \frac{\sqrt{\alpha}\lambda k_2[4-3\alpha+\sqrt{\alpha}\alpha]}{4(1-\alpha)^2(1-\sqrt{\alpha})^2}.
\end{aligned}$$

- Central limit theorem:

$$\lim_{t\to+\infty}\frac{N(t)-\bar{\lambda}t}{\sqrt{t}} \overset{d}{=} \sigma B(1),$$

  where:

  - $\bar{\lambda} = \frac{\lambda}{1-\alpha}$;
  - $\sigma^2 = \frac{\lambda}{(1-\alpha)^3}$;
  - $B(1)$ is a standard Brownian motion;
  - $\overset{d}{=}$ means equality in distribution.

5.2.5 Penalization

The objective function has been penalized in a new way, in order to improve the estimated Hawkes Process robustness and compensate some defaults on real data. Regularization aims to correct the main default observed on the data-set, which is the inaccuracy of times of occurrence. Some events might happen before some others according to the data-set, while it is the opposite in real life. The consequence of this inaccuracy may be that these misreported dates could reinforce the wrong coefficients from triggering matrix.

Parameters are estimated by minimizing the MHP negative log-likelihood. The contribution is that two types of penalization, which are specific to our insurance data, are proposed:

- **Sparsity:** we expect that for couples of events which have no influence on each other, the infectivity matrix has zeros for the corresponding coefficients. That is why we add a sparsity constraint, denoted R_1:

$$R_1(\Theta) = \alpha_1 \Big(\sum_{d=1}^{D} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{(a_{i,j}^d)^2}{2} \Big),$$

  where α_1 is a parameter to control the influence of this regularization and $a_{i,j}^d$ is the causality coefficient between events i and j related to the d^{th} triggering function ;

- **Minimizing the influence of overlapping events:** for some events, the date of occurrence that is in the data-set may be delayed of several days, because of administrative issues. Therefore, we could believe that an event 1 is triggered by an event 2, while event 2 actually occurred after event 1. Therefore we propose to penalize the squared sum of coefficients $a_{i,j}^d$ and $a_{j,i}^d$, in order to correct the influence of overlapping events:

$$R_2(\Theta) = \alpha_2 \Big(\sum_{d=1}^{D} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{(a_{i,j}^d + a_{j,i}^d)^2}{2} \Big),$$

  where α_2 is a parameter to control the influence of this regularization.

To observe the influence of penalization, we compare the list of higher interactions without and with regularization. Especially, we check whether couples of events (Event 1, Event 2) which appear on top of the list with both causations Event 1 $\rightarrow$ Event 2 and Event 2 $\rightarrow$ Event 1 without regularization appear with only one causation with regularization. Table 3.4 compares higher interactions (as defined in Section 3.3) in both cases by highlighting the various couples of events.

Without regularization			**With regularization**		
Triggering	**Triggered**	**Coefficient**	**Triggering**	**Triggered**	**Coefficient**
Move	Household composition	5.90×10^{-2}	Change of vehicle	Insurance cover modification	4.45×10^{-2}
Change of vehicle	Insurance cover modification	5.38×10^{-2}	Move	Household composition	3.07×10^{-2}
Insurance cover modification	Change of vehicle	4.89×10^{-2}	Insurance cover modification	Insurance cover modification	1.46×10^{-2}
Insurance cover modification	Insurance cover modification	4.70×10^{-2}	Change of job	Change of job	1.29×10^{-2}
Change of vehicle	Household composition	2.39×10^{-2}	Change of job	Household composition	1.02×10^{-2}
Change of job	Change of job	2.35×10^{-2}	Change of vehicle	Household composition	8.64×10^{-3}
Household composition	Move	1.91×10^{-2}	Birth	Household composition	5.49×10^{-3}
Move	Birth	1.82×10^{-2}	Household composition	Move	4.23×10^{-3}
Change of job	Household composition	1.35×10^{-2}	Move	Birth	3.61×10^{-3}
Move	Insurance cover modification	8.40×10^{-3}	Change of job	Move	2.86×10^{-3}

Table 5.1: Comparison of highest interactions between events without and with regularization

We notice that with regularization, only one causation from couples of events (Move, Household composition) and (Change of vehicle, Insurance cover modification) is kept as expected. The other causation has a lower coefficient, which allows us to define which event triggers the other for every couple.

5.2.6 Package `[libraryname]` featuring estimation, simulation, plotting

The R package `[libraryname]` implements several functions which allow to complete the following tasks:

- Data preparation;
- Parameters estimation;
- Simulation;
- Plotting: intensity, events history, SHAP values.

There exists no R packages which includes these features. Most of them are either limited to univariate analysis or don't propose an implementation of parameters estimation.

5.3 Perspectives

In this subsection, we present a perspective of future work and improvements associated to the recommendation system.

5.3.1 Pilot phase

The next step is to launch a pilot phase in the same fashion than described in Section 1.7.4.2 in order to test whether the recommendation system works on real customers. Several guidelines need to be clarified before beginning to test the recommendation system:

- Which products/guarantees are targeted by the campaign? This choice could depend on marketing concerns;
- How is the recommendation system proposed to the customers? Agents have the final word on whether the customers should receive the recommendation or not, then they must agree to use the recommendation system for a certain amount of suggestions;
- Which Return On Investment (ROI) should be considered to evaluate the performance of the recommendation system?
  - We should consider the acceptance rate, which is the ratio of accepted suggestions among all the recommendations;
  - We could also evaluate the efficiency of the life events detection by evaluating whether the recommendations triggered by the occurrence of a life event are followed or not. We could observe the results for the different kinds of life events.

5.3.2 Integration to the IT system

Currently generated through the package `[libraryname]`, the predictions from the recommendation should be integrated to the Foyer IT system, so that they are updated on a daily basis and that they are accessible to Foyer employees at any time. The recommendation system should be implemented in a SAS program which should be automatically run every day. Also, the SAS environment includes a module to access interactive dashboards which could be used for the ROI follow-up.

5.3.3 Recommendation system improvements

We list below several propositions to improve the recommendation system built in this thesis.

5.3.3.1 Background intensity

The background intensity, personalized for each customer, is calculated according to one of the following methods, depending on the type of event (see Section 3.2.2):

- Machine Learning algorithm, which takes as an input explanatory variables and learns from this database the background intensity;
- Deterministic estimation from statistical studies;
- Uniform background intensity for events assumed not to depend on customers' profile.

The first and the second method could be improved by including more relevant data and studies. For instance, the learning of the background intensity of the vehicle change could take into account telematics data, so that the algorithm knows the behaviour of the customer on the road and could calculate a probability to change his vehicle in function of his driving habits. This category of data is not collected yet. Concerning the event "Change of job", the background intensity is currently calculated in function of customers' age: we could take into account the socio-professional category of the customer because the turnover rate depends on the job. However, the data quality about this information is not currently sufficient to perform analysis on it.

5.3.3.2 Explainability

Currently, the background intensity explainability is based on SHAP values, which needs to know the conditional distribution between all the subsets of variables. In the current recommendation system, we assume that all the features are independent, except for two variables: claims payments and ALAE, whose dependency is modelled by a copula. In order to improve the method, we could consider a more complex structure by including more dependencies into the dataset. It would imply to observe more precisely the correlations between variables in order to find the most adequate structure.

References

[1] L. Lesage, M. Deaconu, A. Lejay, J. Meira, G. Nichil, and R. State, "A recommendation system for car insurance," *European Actuarial Journal*, no. 10, pp. 377–398, 2020. [Cited on pages i, iii, 10, and 80]

[2] L. Lesage, M. Deaconu, A. Lejay, J. A. Meira, G. Nichil, and R. State, "Hawkes processes framework with a Gamma density as excitation function: application to natural disasters for insurance." working paper or preprint, Dec. 2020. [Cited on pages iii, 10, 51, 53, and 72]

[3] L. Lesage, M. Deaconu, A. Lejay, J. A. Meira, G. Nichil, and R. State, "A Recommendation System For Insurance Built With A Multivariate Hawkes Process Based On Customers' Life Events." working paper or preprint, Dec. 2021. [Cited on pages iii and 10]

[4] L. Lesage, "A Hawkes process to make aware people of the severity of COVID-19 outbreak: application to cases in France," research report, Université de Lorraine ; University of Luxembourg, Mar. 2020. [Cited on pages iii and 41]

[5] S. I. national de la statistique et des études économiques du Grand-Duché de Luxembourg), ed., *Luxembourg in figures - 2020*. 2020. [Cited on page 1]

[6] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, Dec. 2016. [Cited on page 3]

[7] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, (San Francisco, CA, USA), p. 43–52, Morgan Kaufmann Publishers Inc., 1998. [Cited on page 3]

[8] Z.-D. Zhao and M.-s. Shang, "User-based collaborative-filtering recommendation algorithms on hadoop," in *2010 Third International Conference on Knowledge Discovery and Data Mining*, pp. 478–481, 2010. [Cited on pages 3 and 30]

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, (New York, NY, USA), p. 285–295, Association for Computing Machinery, 2001. [Cited on pages 3 and 30]

[10] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, (New York, NY, USA), p. 39–46, Association for Computing Machinery, 2010. [Cited on pages 3 and 30]

[11] J. Ramos, "Using tf-idf to determine word relevance in document queries," 2003. [Cited on page 3]

[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, Feb. 2019. [Cited on page 3]

[13] L. Rokach, G. Shani, B. Shapira, E. Chapnik, and G. Siboni, "Recommending insurance riders," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, (New York, NY, USA), p. 253–260, Association for Computing Machinery, 2013. [Cited on page 4]

[14] T. De Pessemier, J. Minnaert, K. Vanhecke, S. Dooms, and L. Martens, "Social recommendations for events," *CEUR Workshop Proceedings*, vol. 1066, 01 2013. [Cited on page 5]

[15] A. Macedo and L. Marinho, "Event recommendation in event-based social networks," in *1st International Workshop on Social Personalisation*, 09 2014. [Cited on page 5]

[16] W. Zhang and J. Wang, *A Collective Bayesian Poisson Factorization Model for Cold-Start Local Event Recommendation*, p. 1455–1464. New York, NY, USA: Association for Computing Machinery, 2015. [Cited on page 5]

[17] H.-C. Wang, H.-T. Jhou, and Y.-S. Tsai, "Adapting topic map and social influence to the personalized hybrid recommender system," *Information Sciences*, 04 2018. [Cited on page 5]

[18] M. Denuit and A. Charpentier, eds., *Mathématiques de l'assurance non-vie ; Tome 1 : Prinicipe fondamentaux de théorie du risque.* Economica, 2004. [Cited on page 5]

[19] J. Yan, H. Xu, and L. Li, "Modeling and applications for temporal point processes," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, (New York, NY, USA), p. 3227–3228, Association for Computing Machinery, 2019. [Cited on page 6]

[20] H. Xu, W. Wu, S. Nemati, and H. Zha, "Patient flow prediction via discriminative learning of mutually-correcting processes (extended abstract)," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 37–38, 2017. [Cited on page 6]

[21] A. Zammit-Mangion, M. Dewar, V. Kadirkamanathan, and G. Sanguinetti, “Point process modelling of the afghan war diary,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 31, pp. 12414–12419, 2012. [Cited on page 6]

[22] Y. Ogata, “Statistical models for earthquake occurrences and residual analysis for point processes,” *Journal of the American Statistical Association*, vol. 83, no. 401, pp. 9–27, 1988. [Cited on page 6]

[23] U. Bhat, *An Introduction to Queueing Theory: Modeling and Analysis in Applications*. Statistics for Industry and Technology, Birkhäuser Boston, 2015. [Cited on page 6]

[24] R. F. Engle and J. R. Russell, “Autoregressive conditional duration: A new model for irregularly spaced transaction data,” *Econometrica*, vol. 66, no. 5, pp. 1127–1162, 1998. [Cited on page 6]

[25] W. Xiao, X. Xu, K. Liang, J. Mao, and J. Wang, “Job recommendation with hawkes process: An effective solution for recsys challenge 2016,” in *Proceedings of the Recommender Systems Challenge*, RecSys Challenge ’16, (New York, NY, USA), Association for Computing Machinery, 2016. [Cited on page 8]

[26] J. Shang and M. Sun, “Local low-rank hawkes processes for temporal user-item interactions,” in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 427–436, 2018. [Cited on page 8]

[27] N. Du, Y. Wang, N. He, J. Sun, and L. Song, “Time-sensitive recommendation from recurrent user activities,” in *Advances in Neural Information Processing Systems* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, pp. 3492–3500, Curran Associates, Inc., 2015. [Cited on page 8]

[28] G. Zhang, J. Wang, N. Jiang, and Y. Sheng, “A point-of-interest recommendation method based on hawkes process,” *Acta Geodaetica et Cartographica Sinica*, vol. 47, no. 9, p. 1261, 2018. [Cited on page 8]

[29] R. Zaatour, *hawkes: Hawkes process simulation and calibration toolkit*, 2014. R package version 0.0-4. [Cited on pages 9 and 83]

[30] P. F. Halpin, *hawkes: EM estimation of multivariate Hawkes process*, 2021. R package version 0.1.0. [Cited on pages 9, 42, and 83]

[31] F. Cheysson, *hawkesbow: Estimation of Hawkes Processes from Binned Observations*, 2021. R package version 1.0.2. [Cited on pages 9 and 84]

[32] D. S. Harte, “PtProcess: An R package for modelling marked point processes indexed by time,” *Journal of Statistical Software*, vol. 35, no. 8, pp. 1–32, 2010. [Cited on pages 9 and 84]

[33] S. W. Linderman and R. P. Adams, “Scalable bayesian inference for excitatory point process networks,” 2015. [Cited on pages 9 and 84]

[34] E. Bacry, M. Bompaire, S. Gaïffas, and S. Poulsen, “Tick: a python library for statistical learning, with a particular emphasis on time-dependent modelling,” 2018. [Cited on pages 9 and 84]

[35] M. Lutz and E. Biernat, *Data Science : fondamentaux et études de cas*. Eyrolles, 2017. [Cited on page 12]

[36] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Third Edition*. Prentice Hall, 2010. [Cited on page 12]

[37] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012. [Cited on page 12]

[38] C. Zhang and Q. Yang, “Data preparation for data mining.,” *Applied Artificial Intelligence*, vol. 17, pp. 375–381, 05 2003. [Cited on page 13]

[39] M. Kuhn and K. Johnson, *Applied predictive modeling*. New York, NY: Springer, 2013. [Cited on pages 13 and 17]

[40] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O’Reilly Media, Inc., 1st ed., 2018. [Cited on page 15]

[41] J. M. Kanter and K. Veeramachaneni, “Deep feature synthesis: Towards automating data science endeavors,” in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, 2015. [Cited on page 15]

[42] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. [Cited on page 15]

[43] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. [Cited on page 16]

[44] W. Loh, “Classification and regression trees,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, 2011. [Cited on page 17]

[45] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Cited on page 17]

[46] S. Lundberg and S. Lee, “A unified approach to interpreting model predictions,” *CoRR*, vol. abs/1705.07874, 2017. [Cited on pages 19 and 20]

[47] L. S. Shapley, *17. A Value for n-Person Games*, pp. 307–318. Princeton University Press, 2016. [Cited on page 19]

[48] K. Aas, M. Jullum, and A. Løland, "Explaining individual predictions when features are dependent: More accurate approximations to shapley values," *Artificial Intelligence*, vol. 298, p. 103502, 2021. [Cited on pages 20, 21, and 22]

[49] F. Miller, A. Vandome, and M. John, *Mahalanobis Distance*. VDM Publishing, 2010. [Cited on page 21]

[50] Y. Fang and L. Madsen, "Modified gaussian pseudo-copula: Applications in insurance and finance," *Insurance: Mathematics and Economics*, vol. 53, no. 1, pp. 292–301, 2013. [Cited on page 21]

[51] E. W. Frees and E. A. Valdez, "Understanding relationships using copulas," *North American actuarial journal*, vol. 2, no. 1, pp. 1–25, 1998. [Cited on page 21]

[52] M. Denuit, O. Purcaru, and I. Keilegom, "Bivariate archimedean copula models for censored data in non-life insurance," *Journal of Actuarial Practice*, vol. 13, 01 2006. [Cited on page 21]

[53] M. Kendall and J. Gibbons, *Rank Correlation Methods*. A Charles Griffin title, Edward Arnold, 1990. [Cited on page 21]

[54] M. J. Sklar, "Fonctions de répartition à n dimensions et leurs marges," 1959. [Cited on page 21]

[55] T. Nagler, U. Schepsmeier, J. Stoeber, E. C. Brechmann, B. Graeler, and T. Erhardt, *VineCopula: Statistical Inference of Vine Copulas*, 2021. R package version 2.4.3. [Cited on page 21]

[56] W. Wang and M. T. Wells, "Model selection and semiparametric inference for bivariate failure-time data," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 62–72, 2000. [Cited on page 21]

[57] H. Joe, *Multivariate Models and Multivariate Dependence Concepts (1st ed.)*. Chapman and Hall/CRC, 1997. [Cited on page 21]

[58] F. Aldhufairi, R. Samanthi, and J. Sepanski, "New families of bivariate copulas via unit lomax distortion," *Risks*, vol. 8, p. 106, 10 2020. [Cited on page 21]

[59] N. Sellereite, M. Jullum, and A. Redelmeier, *shapr: Prediction Explanation with Dependence-Aware Shapley Values*, 2021. R package version 0.2.0. [Cited on pages 22 and 23]

[60] S. Maksymiuk, A. Gosiewska, and P. Biecek, "Landscape of R packages for explainable artificial intelligence," *CoRR*, vol. abs/2009.13248, 2020. [Cited on page 22]

[61] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB ’94, (San Francisco, CA, USA), p. 487–499, Morgan Kaufmann Publishers Inc., 1994. [Cited on page 26]

[62] T. Kumar-Tummapudi, “Effective navigation of query results using apriori algorithm,” *International Journal of Computer Science and Information Technologies*, vol. 6, pp. 1952 – 1955, 2015. [Cited on page 26]

[63] M. Qazi, G. M. Fung, K. J. Meissner, and E. R. Fontes, “An insurance recommendation system using bayesian networks,” in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys ’17, (New York, NY, USA), p. 274–278, Association for Computing Machinery, 2017. [Cited on page 32]

[64] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes. Vol. I.* Probability and its Applications (New York), New York: Springer-Verlag, second ed., 2003. Elementary theory and methods. [Cited on pages 38 and 39]

[65] J. G. Rasmussen, “Lecture notes: Temporal point processes and the conditional intensity function,” 2018. [Cited on page 38]

[66] A. G. Hawkes, “Spectra of some self-exciting and mutually exciting point processes,” *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971. [Cited on pages 40, 41, and 64]

[67] P. J. Laub, T. Taimre, and P. K. Pollett, “Hawkes processes,” 2015. [Cited on pages 40, 49, and 64]

[68] E. Bacry and J.-F. Muzy, “Second order statistics characterization of hawkes processes and non-parametric estimation,” 2015. [Cited on page 40]

[69] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, “Self-exciting point process modeling of crime,” *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011. [Cited on page 40]

[70] E. Bacry, I. Mastromatteo, and J. Muzy, “Hawkes processes in finance,” *Market Microstructure and Liquidity*, vol. 01, 02 2015. [Cited on pages 40 and 59]

[71] Z. Cheng and Y. Seol, “Diffusion Approximation of a Risk Model with Non-Stationary Hawkes Arrivals of Claims,” *Methodology and Computing in Applied Probability*, vol. 22, pp. 555–571, June 2020. [Cited on page 40]

[72] Y. Bessy-Roland, A. Boumezoued, and C. Hillairet, “Multivariate hawkes process for cyber insurance,” *Annals of Actuarial Science*, vol. 15, pp. 1–26, 06 2020. [Cited on page 40]

[73] P. Babuna, X. Yang, A. Gyilbag, D. A. Awudi, D. Ngmenbelle, and D. Bian, "The impact of covid-19 on the insurance industry," *International Journal of Environmental Research and Public Health*, vol. 17, no. 16, 2020. [Cited on page 41]

[74] A. Agosto and P. Giudici, "A poisson autoregressive model to understand covid-19 contagion dynamics," *SSRN Electronic Journal*, 01 2020. [Cited on page 42]

[75] C. from John Hopkins University, *https://github.com/CSSEGISandData/COVID-19*. 2020. [Cited on page 42]

[76] P. Halpin and P. De Boeck, "Modelling dyadic interaction with hawkes processes," *Psychometrika*, vol. 78, pp. 793–814, 10 2013. [Cited on page 42]

[77] Y. Ogata, "On lewis' simulation method for point processes," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 23–31, 1981. [Cited on pages 44, 49, and 76]

[78] "Coronavirus : les stratégies de dépistage laissent perplexes patients et professionnels." `https://www.lemonde.fr/planete/article/2020/03/11/coronavirus-les-strategies-de-depistage-laissent-perplexes-patients-et-profession` `html`, 2020. [Cited on page 47]

[79] "Coronavirus testing: Criteria and numbers by country." `https://www.worldometers.info/coronavirus/covid-19-testing/`, 2020. [Cited on page 47]

[80] H. Xu, M. Farajtabar, and H. Zha, "Learning granger causality for hawkes processes," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1717–1726, PMLR, 20–22 Jun 2016. [Cited on pages 51 and 72]

[81] K. Zhou, H. Zha, and L. Song, "Learning triggering kernels for multi-dimensional hawkes processes," *30th International Conference on Machine Learning, ICML 2013*, pp. 2338–2346, 01 2013. [Cited on page 54]

[82] S. Asmussen, *Applied Probability and Queues*. United States: IEEE Computer Society Press, 1987. [Cited on page 55]

[83] X. Gao and L. Zhu, "Functional central limit theorems for stationary hawkes processes and application to infinite-server queues.," *Queueing Syst. Theory Appl.*, vol. 90, no. 1-2, pp. 161–206, 2018. [Cited on pages 58 and 59]

[84] E. Bacry, S. Delattre, M. Hoffmann, and J. Muzy, "Some limit theorems for hawkes processes and application to financial statistics," *Stochastic Processes and their Applications*, vol. 123, no. 7, pp. 2475–2499, 2013. A Special Issue on the Occasion of the 2013 International Year of Statistics. [Cited on page 59]

[85] J. J. D'Azzo and C. D. Houpis, *Linear Control System Analysis and Design: Conventional and Modern.* McGraw-Hill Higher Education, 4th ed., 1995. [Cited on page 61]

[86] R. Lemonnier, K. Scaman, and A. Kalogeratos, "Multivariate hawkes processes for large-scale inference," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, Feb. 2017. [Cited on page 70]

[87] S. I. national de la statistique et des études économiques du Grand-Duché de Luxembourg). Available at `https://statistiques.public.lu`, 2021. [Cited on pages 71 and 109]

[88] I. I. N. de la Statistique et des Études Économiques). Available at `https://dares.travail-emploi.gouv.fr/sites/default/files/pdf/2018-049v2.pdf`, 2021. [Cited on pages 71 and 111]

[89] I. I. N. de la Statistique et des Études Économiques). Available at `https://www.insee.fr/fr/statistiques/2882020`, 2021. [Cited on page 71]

[90] E. Bacry and J.-F. Muzy, "First- and Second-Order Statistics Characterization of Hawkes Processes and Non-Parametric Estimation," *IEEE Transactions on Information Theory*, vol. 62, pp. 2184–2202, Apr. 2016. [Cited on page 75]

[91] A. Henningsen and O. Toomet, "maxlik: A package for maximum likelihood estimation in R," *Computational Statistics*, vol. 26, no. 3, pp. 443–458, 2011. [Cited on page 83]

[92] W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson, *shiny: Web Application Framework for R*, 2020. R package version 1.4.0.2. [Cited on page 107]

Appendix A

Interface developed for the pilot phase of the recommendation system

The pilot phase, described in Section 1.7, allowed to test the first part of the recommendation on real customers. The objective was to demonstrate the advantages of a recommendation system to Foyer agents, not only for the accuracy of the suggestions but also for the simplicity of use. Accurate recommendations are useless if agents could not have an easy access to them.

That is why an interface was developed at the same time as the first version of the recommendation system. The objective was to provide agents a tool to see the recommendations, sort them, indicate whether the recommendation was accepted or not and display several statistics about the campaign such as:

- Overall acceptance rate;
- Number of customers contacted;
- Average premiums subscribed among accepted recommendations;
- Total premium earned during the whole campaign;
- Etc.

The tool was developed mostly with R-Shiny [92] and an overview of the main page could be found in Figure A.1.

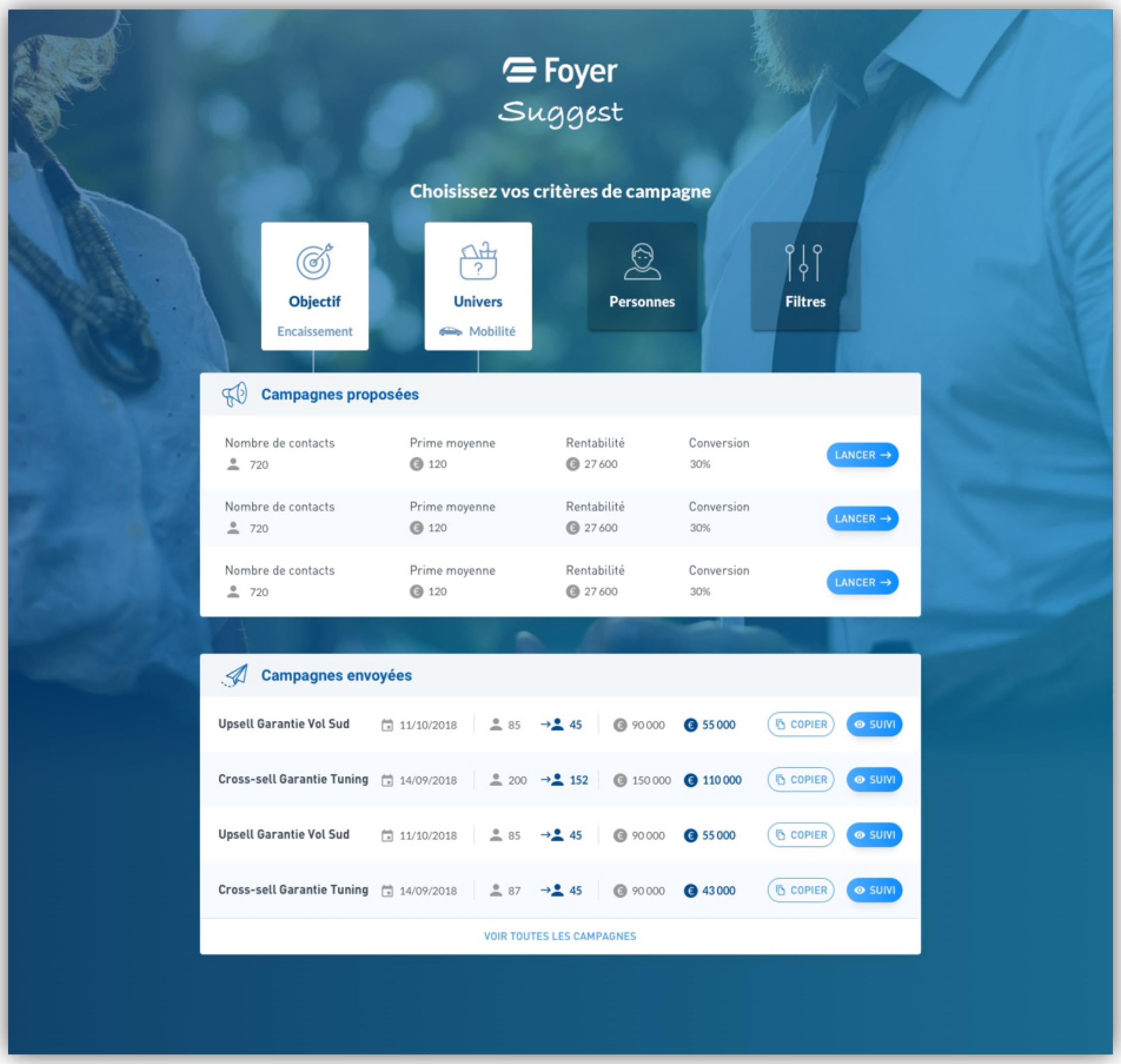


Figure A.1: Main page of the interface developed for the agents on the occasion of the pilot phase

Appendix B

Fertility rates by age groups in Luxembourg in 2015

Table B.1 presents the fertility rate by age groups in Luxembourg observed in 2015, from the STATEC website [87]. It corresponds to the average number of live births per 1000 women per age group.

Age group	**Fertility**
15-19 years	5.6
20-24 years	30.8
25-29 years	77.6
30-34 years	103.1
35-39 years	64.6
40-44 years	12.9
Average	51.6

Table B.1: Fertility rates by age groups in Luxembourg in 2015

Appendix C

Change of job rates by age groups in France, observed from 2010 to 2015

Table C.1 presents the change of job rates by age groups in France observed from 2010 to 2015, from a survey of the French Institute of Statistics (see [88]). It corresponds to the percentage of French people employed in 2010 who change of job between 2010 and 2015. We assume that people under 20 belong to the youngest class and those above 50 to the oldest one.

Age group	**Rate**
20-29 years	33
30-39 years	22
40-50 years	16

Table C.1: Change of job rates by age groups in France observed from 2010 to 2015

Appendix D

Estimation of the infectivity matrix

Tables D.1 to D.6 present the estimation of the infectivity matrix $\boldsymbol{A_d}, d \in \{1, ..., D\}$, denoted by $\hat{\boldsymbol{A}}_{\boldsymbol{d}}$, where $\hat{\boldsymbol{A}}_{\boldsymbol{d}} = \left(\hat{a}^d_{i,j}\right)_{i,j=1,...,m}$.

Triggered \ **Triggering**	Ins. cov. modif.	Change of vehicle	House. compo.	Birth	Change of job	Move
Insurance cover modif.	3.6×10^{-4}	4.3×10^{-2}	5.5×10^{-4}	2.6×10^{-10}	7.4×10^{-4}	3.1×10^{-5}
Change of vehicle	5.9×10^{-16}	2.0×10^{-17}	4.8×10^{-10}	3.2×10^{-9}	5.5×10^{-9}	2.0×10^{-13}
Household composition	4.8×10^{-13}	2.3×10^{-8}	0.0	7.7×10^{-8}	1.5×10^{-7}	2.2×10^{-5}
Birth	2.7×10^{-18}	4.5×10^{-8}	2.9×10^{-5}	0.0	1.6×10^{-4}	1.4×10^{-4}
Change of job	8.4×10^{-5}	1.7×10^{-9}	1.0×10^{-4}	1.5×10^{-8}	3.6×10^{-4}	7.0×10^{-6}
Move	9.2×10^{-9}	1.4×10^{-12}	2.2×10^{-3}	5.0×10^{-10}	3.4×10^{-9}	0.0

Table D.1: Estimation of the infectivity matrix $\boldsymbol{A}_1$

Triggered \ **Triggering**	Ins. cov. modif.	Change of vehicle	House. compo.	Birth	Change of job	Move
Insurance cover modif.	1.5×10^{-3}	1.0×10^{-3}	2.6×10^{-4}	1.6×10^{-6}	8.6×10^{-4}	1.6×10^{-3}
Change of vehicle	2.3×10^{-16}	5.3×10^{-16}	1.4×10^{-5}	7.0×10^{-5}	4.7×10^{-4}	5.9×10^{-8}
Household composition	1.5×10^{-4}	1.7×10^{-3}	0.0	9.9×10^{-4}	2.2×10^{-3}	7.4×10^{-3}
Birth	3.9×10^{-7}	5.6×10^{-5}	7.3×10^{-11}	0.0	7.4×10^{-4}	2.0×10^{-3}
Change of job	2.5×10^{-4}	3.6×10^{-5}	2.3×10^{-4}	3.1×10^{-5}	2.8×10^{-3}	3.7×10^{-4}
Move	8.1×10^{-5}	9.5×10^{-8}	1.5×10^{-3}	3.0×10^{-5}	1.2×10^{-3}	0.0

Table D.2: Estimation of the infectivity matrix $\boldsymbol{A}_2$

Triggered \ **Triggering**	Ins. cov. modif.	Change of vehicle	House. compo.	Birth	Change of job	Move
Insurance cover modif.	2.1×10^{-3}	0.0	4.8×10^{-9}	1.0×10^{-6}	8.0×10^{-5}	5.0×10^{-6}
Change of vehicle	1.9×10^{-1}	4.6×10^{-16}	5.5×10^{-7}	9.9×10^{-5}	2.1×10^{-4}	1.1×10^{-8}
Household composition	4.1×10^{-4}	2.1×10^{-3}	0.0	1.1×10^{-3}	2.1×10^{-3}	7.0×10^{-3}
Birth	9.6×10^{-6}	9.9×10^{-6}	7.7×10^{-11}	0.0	5.4×10^{-4}	6.5×10^{-4}
Change of job	2.1×10^{-4}	9.7×10^{-5}	2.1×10^{-4}	5.6×10^{-5}	2.5×10^{-3}	6.8×10^{-4}
Move	2.1×10^{-4}	1.7×10^{-7}	3.0×10^{-4}	1.2×10^{-4}	6.6×10^{-4}	0.0

Table D.3: Estimation of the infectivity matrix $\boldsymbol{A}_3$

Triggered \ **Triggering**	Ins. cov. modif.	Change of vehicle	House. compo.	Birth	Change of job	Move
Insurance cover modif.	2.7×10^{-3}	0.0	2.7×10^{-10}	6.4×10^{-7}	1.9×10^{-4}	7.6×10^{-10}
Change of vehicle	2.0×10^{-16}	9.2×10^{-16}	1.8×10^{-7}	2.1×10^{-4}	2.0×10^{-4}	1.7×10^{-9}
Household composition	5.6×10^{-4}	1.9×10^{-3}	0.0	8.1×10^{-4}	2.1×10^{-3}	6.1×10^{-3}
Birth	2.1×10^{-5}	1.7×10^{-8}	1.1×10^{-6}	4.6×10^{-23}	1.9×10^{-5}	3.8×10^{-5}
Change of job	1.7×10^{-4}	3.4×10^{-5}	5.6×10^{-6}	2.7×10^{-4}	2.3×10^{-3}	3.7×10^{-4}
Move	2.3×10^{-4}	1.9×10^{-7}	1.6×10^{-5}	1.9×10^{-4}	5.7×10^{-4}	0.0

Table D.4: Estimation of the infectivity matrix $\boldsymbol{A}_4$

Triggering / **Triggered**	Ins. cov. modif.	Change of vehicle	House. compo.	Birth	Change of job	Move
Insurance cover modif.	3.3×10^{-3}	0.0	2.6×10^{-14}	8.0×10^{-8}	3.1×10^{-4}	1.7×10^{-9}
Change of vehicle	2.2×10^{-16}	1.4×10^{-15}	5.0×10^{-8}	5.6×10^{-5}	1.4×10^{-4}	1.9×10^{-10}
Household composition	6.6×10^{-4}	1.4×10^{-3}	4.1×10^{-26}	1.1×10^{-3}	1.9×10^{-3}	5.1×10^{-3}
Birth	3.0×10^{-5}	9.1×10^{-7}	2.2×10^{-12}	2.9×10^{-18}	1.0×10^{-5}	1.4×10^{-4}
Change of job	1.4×10^{-4}	1.9×10^{-5}	9.5×10^{-8}	3.1×10^{-4}	2.0×10^{-3}	4.7×10^{-4}
Move	2.1×10^{-4}	1.5×10^{-7}	6.8×10^{-5}	1.2×10^{-4}	9.5×10^{-5}	3.1×10^{-19}

Table D.5: Estimation of the infectivity matrix $\boldsymbol{A}_5$

Triggering / **Triggered**	Ins. cov. modif.	Change of vehicle	House. compo.	Birth	Change of job	Move
Insurance cover modif.	3.7×10^{-3}	0.0	5.9×10^{-12}	1.3×10^{-8}	1.9×10^{-6}	8.1×10^{-11}
Change of vehicle	2.4×10^{-16}	2.9×10^{-15}	5.0×10^{-10}	4.1×10^{-5}	6.6×10^{-5}	9.2×10^{-11}
Household composition	6.9×10^{-4}	1.3×10^{-3}	5.7×10^{-25}	1.2×10^{-3}	1.4×10^{-3}	4.1×10^{-3}
Birth	3.4×10^{-5}	2.3×10^{-7}	2.4×10^{-15}	3.8×10^{-15}	1.0×10^{-5}	5.1×10^{-4}
Change of job	1.2×10^{-4}	1.1×10^{-5}	3.7×10^{-6}	3.0×10^{-4}	1.8×10^{-3}	9.3×10^{-5}
Move	2.1×10^{-4}	2.5×10^{-8}	1.7×10^{-6}	1.1×10^{-5}	1.7×10^{-4}	1.8×10^{-18}

Table D.6: Estimation of the infectivity matrix $\boldsymbol{A}_6$