EPFL

Thèse n° 10 991

Self-Exciting Point Processes: Identification and Control

Présentée le 21 janvier 2022

Collège du management de la technologie
Chaire en opérations, économie et stratégie
Programme doctoral en management de la technologie

pour l'obtention du grade de Docteur ès Sciences

par

Michael MARK

Acceptée sur proposition du jury

Prof. D. Kuhn, président du jury
Prof. T. A. Weber, directeur de thèse
Prof. A. G. Hawkes, rapporteur
Prof. D. Sornette, rapporteur
Prof. D. Challet, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

Dedicated to my parents for their unwavering love and support.

Acknowledgements

Pursuing a doctorate was a spectacular journey that would have been utterly impossible without the support of countless people. First and foremost, I owe my sincerest gratitude to my supervisor, Professor Thomas A. Weber, for giving me the remarkable intellectual and social opportunity to pursue a doctorate at EPFL. I benefited enormously from his vast knowledge, his research intuition, and countless hours of research discussions that had a profound impact on my thinking. Furthermore, I am deeply thankful for the intellectual freedom I enjoyed in choosing my research topics and the support I always knew I had.

Secondly, I would like to express my profound appreciation to my defense committee: Professor Alan Geoffrey Hawkes, Professor Didier Sornette, Professor Damien Challet, and Professor Daniel Kuhn, all of whom were the imaginary giants on whose shoulders I could stand on. Consequently, I am deeply grateful to my coauthors Professor Naveed Chehrazi and Jan Sila, without whom this thesis would not exist. I benefited tremendously from Prof. Chehrazi's fresh perspective on reinforcement learning, and I wish to thank him for bringing more rigor to my work. I thank Jan Sila for being a great colleague, motivator, and above all, true friend. Our long research discussions have been a great inspiration for many of the ideas presented in this thesis.

Additionally, I cannot begin to express my thanks to my friends who accompanied me every step of the way. I thank my lifetime friends Robin and Martin for their friendship, kindness, and all the unforgettable experiences we have shared. I thank my close friend and colleague Rebekah for the special friendship we built over the past four years – you made all the valleys of the doctorate more bearable. Last but not least, a special thanks goes to all my office mates: to Raphael for making me feel that even my worst days are exquisite compared to his, to Max for teaching me the value of research lunches, to George for reminding me that one must not forget to have fun, and finally to Charles for being a good sport and being my Ph.D. student role model.

Finally, I am deeply grateful to my family, who always supported me and shared with me all the ups and downs of my doctorate. I thank my parents for all the opportunities I was given by virtue of their generosity and the many sacrifices they were ready to make, and my sisters Marta and Michaela for raising me and all the fond memories I have of them.

Ultimately, I will forever be indebted to Tereza, my love and my best friend, for her endless support and, above all, the patience she has shown me throughout these years. Thank you for sticking with me and making me a better person.

Abstract

This thesis addresses theoretical and practical aspects of identification and subsequent control of self-exciting point processes. The main contributions correspond to four separate scientific papers.

In the first paper, we address the challenge of robust identification of controlled Hawkes processes in applications with sparsely available data. Specifically, we propose an alternative approach based on an expectation-maximization algorithm, which instrumentalizes the internal branching structure of the process, thus improving the estimator's convergence behavior. Additionally, we show that our method provides a tight lower bound for maximum-likelihood estimates. The relevance of the proposed technique is demonstrated on the practical application of credit collections and trading in the presence of macroeconomic news.

The second and third paper focus on the optimal control of self-exciting point processes using the reinforcement-learning paradigm. Contrary to traditional reinforcement learning applications, environments driven by Hawkes-like dynamics feature an asynchronous action-reward relationship which complicates attributing actions to their consequent rewards, and thus hinders learning. To this end, we formulate a novel reward shaping theorem that provides a continuous reward analogue that enables learning in such environments. Furthermore, with the growing need for interpretable machine-learning models we formulate a monotonicity regularizer that embeds domain expertise into the learning. Our formulation overcomes the challenge of learning interpretable policies by constraining the policy space with a priori expected structural properties, producing state-feedback control laws that can be readily understood and implemented by human decision-makers. Again the results are developed in the context of credit collections but are straightforwardly applicable to other problems with self-exciting dynamics.

Finally, the last paper consists of an empirical investigation of cryptocurrency market microstructure through the optics of Hawkes processes. We construct a 'reflexivity' index that measures the activity generated endogenously within cryptocurrency markets by fitting a univariate self-exciting Hawkes process with two classes of parametric kernels to high-frequency trading data. Our parsimonious model allows for an elegant separation and quantification of endogenous and exogenous dynamics, and thus allows for a direct market microstructure comparison with traditional asset classes in terms of identified branching ratios. Furthermore, we formulate a 'Hawkes disorder problem,' as a generalization of the established Poisson disorder problem, and provide a simulation-based approach to determining an optimal observation horizon—a critical consideration in the high-frequency finance context. Our analysis

suggests that Bitcoin mid-price dynamics feature long-memory properties, well explained by the power-law kernel, at a level of criticality similar to fiat-currency markets.

Key words

Self-exciting point process, Hawkes process, identification, branching structure, reinforcement learning, monotonicity regularization, market microstructure, cryptoassets.

Résumé

Cette thèse aborde les aspects théoriques et pratiques de l'identification et du contrôle optimal des processus ponctuels auto-excitant. Les contributions principales correspondent à quatre articles scientifiques distincts.

Dans le premier article, nous abordons le défi de l'identification robuste des processus de Hawkes contrôlés dans des applications avec des données éparses. Plus précisément, nous proposons une nouvelle approche basée sur un algorithme d'espérance-maximisation, qui utilise la structure interne du processus pour améliorer les qualités de convergence de l'estimateur. De plus, nous montrons que notre méthode produit une borne inférieure stricte pour les estimations de vraisemblance maximale. La pertinence de la technique proposée est démontrée sur l'application pratique des recouvrements de crédit et du trading en présence d'évènements macroéconomiques.

Les deuxième et troisième articles se concentrent sur le contrôle optimal des processus ponctuels auto-excitant en utilisant le paradigme d'apprentissage par renforcement. Contrairement aux applications traditionnelles d'apprentissage par renforcement, les environnements pilotés par une dynamique de type Hawkes présentent une relation action-récompense asynchrone qui complique l'attribution des actions à leurs récompenses. C'est pourquoi nous formulons un nouveau théorème de transformation de récompense (reward shaping) qui permet l'apprentissage dans de tels environnements. De plus, avec le besoin croissant de modèles d'apprentissage automatique interprétables, nous proposons une méthode de régularisation de monotonie qui intègre l'expertise du domaine dans l'apprentissage. Notre formulation surmonte le défi de l'apprentissage de politiques interprétables en contraignant l'espace des stratégies faisable en utilisant des propriétés structurelles, produisant des lois de contrôle par rétroaction d'état qui peuvent être facilement comprises et mises en pratique par les décideurs humains. Pour ces deux articles aussi, les résultats sont testés dans le contexte des recouvrements de crédit, mais ils sont également applicables à d'autres problèmes avec une dynamique d'auto-excitation.

Enfin, le dernier article consiste en une enquête empirique sur la microstructure du marché des crypto-monnaies à travers l'optique des processus de Hawkes. Nous construisons un indice de réflexivité qui mesure l'activité générée de manière endogène sur les marchés de la crypto-monnaie en adaptant un processus de Hawkes auto-excitant univarié avec deux classes de noyaux paramétriques aux données de trading à haute fréquence. Notre modèle détaillé permet une séparation et une quantification élégantes des dynamiques endogènes et exogènes, et permet ainsi une comparaison directe de la microstructure du marché avec les

classes d'actifs traditionnelles en termes de ratios de branchement identifiés. En outre, nous formulons un « problème du trouble de Hawkes », en tant que généralisation du problème du trouble de Poisson établi, et proposons une approche basée sur la simulation pour déterminer un horizon d'observation optimal — une considération critique dans le contexte financier à haute fréquence. Notre analyse suggère que la dynamique des prix moyens du Bitcoin présente des propriétés de mémoire longue, bien expliquées par le noyau de la loi de puissance, à un niveau de criticité similaire aux marchés de la monnaie fiduciaire.

Mots-clés

Processus de Hawkes, identification, structure de branchement, apprentissage par renforcement, régularisation de monotonie, microstructure du marché, cryptomonnaies.

Contents

Introduction

Motivation

Real-world systems are rarely memoryless. Observable events such as a lightening strike or a company default principally do not arise as a result of some single invisible force, but rather as a consequence of previous smaller events and their interactions developed into a complex network of mutual interdependencies. As a result, when it comes to modeling real event-data streams a memoryless Poisson point process is often of limited applicability as it is unable to capture the underlying causal structure of events (see Fig. 1). Hawkes processes, named after their originator Alan Geoffrey Hawkes, elegantly address this issue by assuming arrival rate that is history-dependent. Typically, an event induces a jump in the intensity which is then dissipated through a memory kernel. Consequently, each jump temporarily influences the probability of additional events occurring which causes arrivals to cluster in time. Not surprisingly, the self-exciting (or more generally self-modulating) mechanism proves to be descriptive for many natural and social phenomena. For instance, at the outset of a pandemic susceptible individuals contracting the disease become carriers that can potentially spread the disease further, hence giving rise to the self-exciting dynamics. The central aim of this thesis is to identify such systems (in terms of some parametrization), and subsequently devise an optimal control schedule that exogenously intervenes in the system intensity in order to attain some prespecified objective. In the example outlined, a decision maker playing the role of a government attempts to contain the spread via costly interventions such as prescribing spread-mitigating measures (e.g., wearing masks or ordering a curfew) which temporarily decrease the rate of spread. A successful epidemic management then boils down to a correct identification of the system dynamics and subsequent design of an optimal policy that minimizes the economic costs incurred by the pandemic. Given the ubiquity of systems featuring a self-exciting feedback loop, the methods developed in this thesis are applicable across a wide spectrum of fields, such as optimal credit-collections or high-frequency market microstructure modelling, both discussed in the main text.

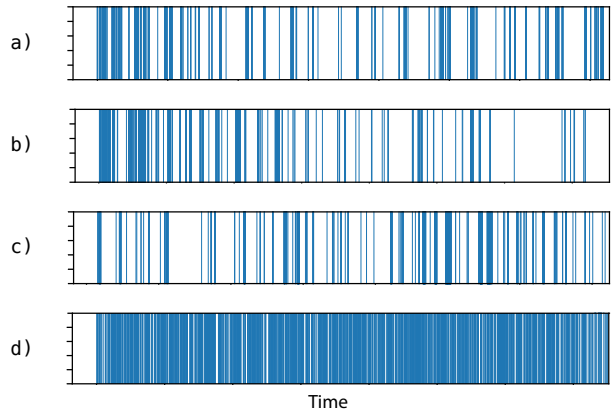


Figure 1: Human and system interactions are often not adequately described by Poisson (memoryless) point processes. Therefore, Hawkes processes serve as an appropriate platform for modeling processes exhibiting such characteristics. a) Snapshot of market orders in the DAX future contract (1073 events) b) First hour of a twitter cascade of @BarrackObama tweet (1141 events) c) Synthetic exponential Hawkes process (1371 events) d) Poisson process (1138 events).

Contribution to the Literature

Nowadays, term *Hawkes process* is used to describe a rather large class of self- and mutually-exciting point processes that goes beyond the original linear specification (Hawkes, 1971a,b). Despite their many desirable properties (e.g., intuitive interpretation of parameters through the branching representation (Hawkes and Oakes, 1974)) and their suitability to capture stochastic clustering, a phenomenon pervasive in many fields, Hawkes processes long had very little practical impact. The first significant application likely appeared in seismology (Vere-Jones, 1975; Adamopoulos, 1976) where the self-exciting mechanism was adopted to model temporally clustered earthquake and after-shock sequences. A decade later, Ogata (1988) formulated a more general spatio-temporal version, the Epidemic Type Aftershock Sequence (ETAS) model, which ignited a steady stream of publications on its many variants (Ogata and Zhuang, 2006). Gradually, Hawkes processes proliferated to other seemingly unrelated disciplines ranging from neuroscience (firing of neurons), social media (tweet cascades), and criminology (terrorist threat modeling) to finance (Truccolo et al., 2005; Truccolo, 2016; Mohler et al., 2011). Finance especially has been a fertile ground for new applications (Bacry et al., 2015) that fuelled further theoretical developments, e.g., the development of new nonparametric estimation methods (Bacry et al., 2016). Owing to the process popularity, the problem of parameter estimation is well explored with classical approaches being direct maximization of the maximum-likelihood function (Rubin, 1972; Ogata, 1978), expectation-maximization of the branching structure augmented maximum-likelihood function (Veen and Schoenberg, 2008), or more seldomly a generalized method of moments (Besbes et al., 2010; Chehrazi and Weber, 2015). On the other hand, the control problem has so far received considerably less

attention. While the theory on the optimal control of Markov jump-diffusion processes and Markov jump processes is well developed, the control problem for Hawkes processes has been largely understudied. Chehrazi et al. (2019) study a specific version of this problem in the context of credit collections. Therein, a debt holder pays off his outstanding balance via a stream of repayments modeled as a marked exponential Hawkes process, while the collector influences the debtor's intensity through costly account treatments in order to maximize the net present value of the amount collected. The authors provide a semi-analytical recursive solution for the optimal value function and characterize the optimal policy via action and inaction regions in the state space. Recent advances in reinforcement learning solicited a more data-driven approach where the intensity dynamics are not pre-specified but rather encoded in a recurrent neural network (Upadhyay et al., 2018). The policy is then represented as a parametric intensity from which action events are sampled, and is learned via an adapted policy gradient algorithm. Despite its many advantages, model-free reinforcement learning is infamous for the inherent hyper-parameter fine tuning, further exacerbated by the use of parameter-rich recurrent networks, and the excessive amounts of data required in order to obtain a high-confidence fit. As a result, opting for this approach can, in many instances, be impractical. Interestingly, the control problem is qualitatively similar to the work of Bayraktar and Ludkovski (2009) who track the state of a hidden Markov jump process by observing a marked Poisson process with local characteristics (i.e., its intensity and mark distributions) depending on the latent state. While the latent (true) intensity is unknown, authors compute a Bayesian estimate of the unobservable intensity which turns out to feature the self-exciting property. That is, if an observable event occurs a positive jump in the Baysian intensity estimate is induced reflecting a confidence increase of being in a high state, while over the periods of inactivity the Bayesian estimate exponentially decays.

This thesis adds to the many recent advances in the field and extends the available theory of identification and control of complex systems with underlying self-exciting dynamics. Specifically, without loss of generality, we focus on linear controlled Hawkes processes with intensity

$$\lambda(t|\mathscr{H}_t) = \mu(t) + \sum_{i:\tau_i<t} g(t-\tau_i, m_i) + a(t), \tag{1}$$

where $\mu(t)$ represents a deterministic time-dependent base rate, $g(t,m)$ specifies the triggering kernel determining the covariance properties of the process, and an open loop -control term $a(t)$. The σ-algebra $\mathscr{H}_t$ describes all the available information up to time t, i.e., $\mathscr{H}_t \triangleq \{(\tau_i, m_i) : \tau_i < t\}$,where τ_i and m_i denote arrival times and their respective marks.

The orchestration of our approach is depicted in Fig. 2. Firstly, a parametric dynamic system featuring Eq. (1) is fitted to an observed data stream consisting of marked temporal events and an open-loop control variable.[1] The dynamic system then repeatedly samples independent

[1] The control variable can always be set to zero returning linear self-exciting dynamics.

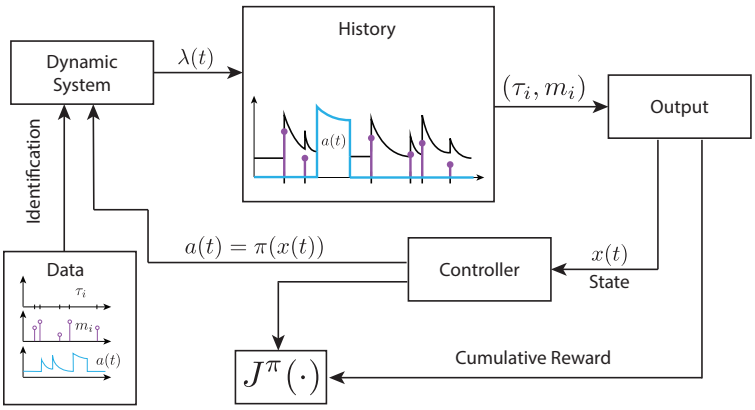


Figure 2: Schema of the identification and control processes.

history realizations that are transformed into a reward signal that encodes decision maker's preferences over realized histories. Taking in the signal and the information about the current state $x(t)$, the controller provides the dynamic system with a new policy $a(t)$ that maximizes some performance metric $J^\pi(\cdot)$, thus finishing a single iteration of the process. The nature of the solution to the optimal control problem is very much determined by the specific structure of the action costs. In situations where action entails a fixed cost infrequent impulse-control interventions become optimal (e.g., in the inventory management it is optimal to do nothing until the stock drops below some trigger level, at which point it is best to replenish). Consequently, the optimal action schedule can be broadly characterized by three disjoint regions in the state space (inaction, action, and holding/transient) (Stokey, 2008).

Thesis Overview

This thesis comprises published, submitted and unpublished articles jointly written with my supervisor Prof. Thomas Alois Weber, and my collaborators Naveed Chehrazi and Jan Sila. The thesis contains four chapters divided into three parts. Chapter 1 addresses the challenge of identification of controlled point processes, specifically in environments with sparse amounts of data. Chapter 2 and Chapter 3 are dedicated to a model-free reinforcement learning approach to decision problems with Hawkes-like asynchronous dynamics. Finally, Chapter 4 contains an empirical investigation of high-frequency market dynamics that adds to the existing discussion on the endo-exo problem from a new angle of crypto currencies.

The chapters are self-contained and organized as follows:

1. **Robust Estimation of Controlled Hawkes Processes**

   The identification of Hawkes-like processes can pose significant challenges. Despite substantial amounts of data, standard estimation methods show significant bias or fail to converge. To overcome these issues, we propose an alternative approach based on an

expectation-maximization algorithm, which instrumentalizes the internal branching structure of the process, thus improving convergence behavior. Furthermore, we show that our method provides a tight lower bound for maximum-likelihood estimates. The approach is discussed in the context of a practical application, namely the collection of outstanding unsecured consumer debt.

2. **Reinforcement Learning Approach to Credit Collections**

   This paper develops a dynamic reinforcement-learning agent capable of finding high-quality policies for the practice of debt collections. At its core, the agent effectively learns how to control a stochastic self-exciting point process in order to maximize an asynchronously obtained reward. To this end, we formulate a stochastic reward shaping theorem that transforms otherwise discretely observed reward into its continuous analogue, thus enabling learning. Because we use a general formulation of the problem as an agent-environment interaction our results are readily extensible beyond the presented application to other problems featuring dynamics based on self-exciting point processes. Furthermore, with the growing need for interpretable machine-learning models we augment the learning procedure with a domaine knowledge regularizer which makes learned policies and value functions intuitively understandable for human decision makers. Finally, we demonstrate the viability of our approach on a traditional neural net approximator as well as on a simpler, linear B-Spline q-function approximator.

3. **Domain-Knowledge Enhanced Policy Gradient: Application to Credit Collections**

   We develop a deterministic policy gradient method that allows for a natural integration of domain expertise into the learning procedure. Domain knowledge can often be formulated in terms of policy monotonicity and/or convexity with respect to relevant state inputs. We augment the standard actor-critic policy approximator using a monotonically regularized loss function that directly integrates domain expertise into the learning. Our formulation overcomes the challenge of learning interpretable policies by constraining the policy space with a priori expected structural properties, producing state-feedback control laws that can be readily understood and implemented by human decision makers. We apply our domain-knowledge enhanced learning approach to the problem of optimal credit collections that features a controlled Hawkes process and an asynchronous action-feedback relationship.

4. **Quantifying Endogeneity of Cryptocurrency Markets**

   We construct a 'reflexivity' index to measure the activity generated endogenously within a market for cryptocurrencies. For this purpose, we fit a univariate self-exciting Hawkes process with two classes of parametric kernels to high-frequency trading data. A parsimonious model of both endogenous and exogenous dynamics enables a direct comparison with exchanges for traditional asset classes, in terms of identified branching ratios. We also formulate a 'Hawkes disorder problem,' as generalization of the established Poisson disorder problem, and provide a simulation-based approach to determining an

optimal observation horizon. Our analysis suggests that Bitcoin mid-price dynamics feature long-memory properties, well explained by the power-law kernel, at a level of criticality similar to fiat-currency markets.

Identification Part I

1 Robust Estimation of Controlled Hawkes Processes

This chapter is based on Mark, M. and Weber, T. A. (2020). Robust Identification of Controlled Hawkes Processes. *Physical Review E*, 101(4):043305.

1.1 Introduction

In contrast to the exogenous intensity of an inhomogeneous Poisson point process, the intensity of a Hawkes process is self-exciting: it depends endogenously on the arrival history (Hawkes, 1971a,b). Any arrival event induces an intensity jump which dissipates through a memory kernel, and this in turn influences the probability of the next arrival event. The first applications of Hawkes processes appeared in seismology, for the analysis of earthquakes and associated aftershock sequences (Hawkes and Adamopoulos, 1973). Since then, self-exciting processes have proved useful across numerous other fields, including finance (Bacry et al., 2015), marketing (Xu et al., 2014), and neuroscience (Truccolo, 2016), to name a few. The performance of the underlying parametric models depends first and foremost on a correct model specification. Here we focus on the identification of a class of linear controlled marked Hawkes processes, where the arrival events include scalar marks and the arrival intensity is regulated by an impulse control. This class of controlled self-exciting processes was first considered by Chehrazi and Weber (2015) to predict the repayment behavior of unsecured loans placed in credit collections. In this application, the collector disposes of a set of account-treatment actions (e.g., establishing first-party contact or sending a notice letter) to exert pressure on the debtor. A similar class of processes was used by Rambaldi et al. (2015) to model foreign-exchange price dynamics subject to exogenous deterministic jumps in the form of news about macroeconomic events.

In the credit-collection example, a misspecification of model parameters leads to faulty predictions of account values and suboptimal account-treatment schedules.[1] Although standard identification techniques, such as maximum-likelihood estimation (MLE), may well be asymptotically consistent, the corresponding estimators tend to exhibit a significant bias as soon

[1] For optimal closed-loop control of repayment processes, see Chehrazi et al. (2019).

as the amount of available data is sparse. This is the case in many practical applications such as credit collections where a delinquent account over the collection history usually features only a few repayment events. In addition, it is often difficult to compute the best-fit parameters because of nonconcavities and near-vanishing gradients of the objective function that lead to ill-conditioned iterations. To ameliorate convergence behavior and estimation performance of standard MLE-methods, we propose a robust estimation method based on an expectation-maximization (EM) algorithm. The latter exploits the branching structure of the process, featuring a primal-dual type approximation. In each iteration, first the lower bound for the likelihood function is updated ("expectation step") before the parameter estimate is reoptimized ("maximization step"). Using a fairly generic setup (in the context of credit collections, to fix ideas), we show that the EM-algorithm achieves substantial improvements in convergence behavior and thus an increased robustness with respect to a broad range of starting values for the parameter vector.

1.1.1 Literature

Due to its relative simplicity, MLE is a common inference method for point processes specified via conditional intensity. A semi-closed form for the estimator was derived by Rubin (1972) who established a link between the conditional density function of the interarrival times and the intensity for regular point processes.[2] The performance of MLE was tested for the first time on seismic data by Ozaki (1979), who also introduced computationally efficient recursive simplification for MLE and derived the Jacobian and Hessian of the corresponding likelihood function. Determining the MLE-estimator then amounts to solving a nonconvex program, using appropriate optimization machinery—with the Jacobian and Hessian readily available for the univariate case. Although Ogata (1978) proved that the associated MLE-estimator is asymptotically normal, efficient, and consistent, the amount of data available for fitting is often insufficient for attaining the asymptotic regime. The resulting estimates tend to be heavily biased or worse, the estimator fails to converge, in many practical applications. For example, such convergence issues were noted in the case of our class of controlled Hawkes processes by Chehrazi and Weber (2015) who proceeded, somewhat *ad hoc*, to filter out unlikely local minima using a Cramér-von Mises goodness-of-fit criterion. The generically poor and unreliable convergence of the MLE-estimator is further exacerbated by the log-likelihood function's exhibiting frequently multimodal or extremely flat behavior near its critical points, resulting overall in a lack of apparent well-posedness (Hadamard, 1902), in the sense that close initialization values can produce very different estimation results. Veen and Schoenberg (2008) documented these anomalies for the popular seismological spatial-temporal Epidemic Type Aftershock Sequence (ETAS) model (Ogata, 1998, 1988, 1993) highlighting the low curvature of the ETAS log-likelihood which deteriorates the performance of the numerical optimization routine. Furthermore, multimodality of the log-likelihood has been empirically confirmed, since for different starting values the optimizer converges generically to different local minima.

[2] A regular point process, defined on a standard probability space $(\Omega, \mathscr{F}_t, \mathbb{P})$, is nonexplosive (i.e., $N(t) < \infty$ for all finite $t \geq 0$).

To overcome the associated computational challenges, the authors suggested to take advantage of the natural branching structure of the process (Hawkes and Oakes, 1974) framing the estimation as an incomplete-data problem where the information about which event triggers other events is unobservable. Building on the case presented by Veen and Schoenberg for the ETAS model, we develop an adapted version of the expectation-maximization algorithm suited for our class of controlled Hawkes processes. Furthermore, we improve the original method by an additional term that shifts the EM-objective function (i.e., the "expected complete log-likelihood," see Section 1.3.2) such that, at the optimum, it becomes a tight lower bound to the log-likelihood function.

1.1.2 Outline

The remainder of this paper is organized as follows. In Section 1.2, we introduce the class of linear controlled Hawkes processes and showcase its importance in two practical examples. Section 1.3 first reviews the MLE-estimator pinpointing its shortcomings and then constructs our estimation method based on the EM-algorithm. In Section 1.4, we compare the two methods in terms of their respective convergence stability and bias. Section 1.5 concludes.

1.2 Controlled Hawkes Processes

1.2.1 Definition

The intensity of a (linear) *controlled Hawkes process* (CHP) is given by

$$\lambda(t|\mathscr{H}_t) = \mu(t) + \sum_{i:\tau_i<t} g(t-\tau_i, m_i) + a(t), \tag{1.1}$$

where $\tau_i \geq 0$ denotes the i-th arrival time and $m_i \in \mathbb{R}$ the corresponding mark, for $i \geq 1$. The background intensity rate $\mu(t)$ is a deterministic function of time $t \geq 0$, the function $g : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$ is a (nonnegative-valued) memory kernel, and the (open-loop) control $a(t)$ is assumed to be a right-continuous function of the form

$$a(t) = \sum_{j:\vartheta_j<t} \Phi_j(t-\vartheta_j), \tag{1.2}$$

where each $\Phi_j : \mathbb{R}_+ \to \mathbb{R}_+$ denotes a (nonnegative-valued) *exogenous kernel* and each ϑ_j an instant at which the control variable a undergoes a jump, for $j \geq 1$. The corresponding control impulses are dissipated via the (nonnegative-valued) exogenous kernels Φ_j as opposed to the *endogenous kernel* g which governs the memory from self-excitation. We assume that on any finite time interval $[0,t]$ the number of impulses is finite, and the intervention times ϑ_j are known in advance. We also assume that both types of kernels satisfy the usual stationarity condition, so $\int_0^\infty \max\{\Phi_j(t), g(t)\}\,dt \leq 1$ for all relevant j.[3] The σ-algebra $\mathscr{H}_t =$

[3] In applications with a finite observation horizon, it is sufficient to impose $\int_0^\infty \max\{\Phi_j(t), g(t)\}\,dt < \infty$.

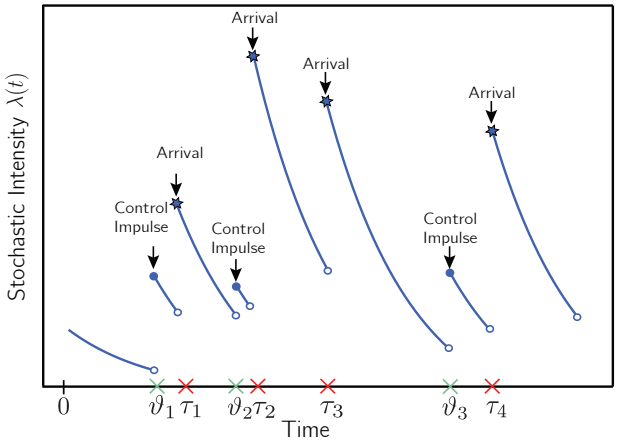


Figure 1.1: Dependence of the stochastic intensity $\lambda(t)$ on arrival history and control impulses.

$\{\{(\tau_i, m_i)\} \times \{\vartheta_j\} : \tau_i < t, \vartheta_j < t\}$ describes the process history, including mark sizes m_i and impulse times ϑ_j. A sample intensity path is shown in Fig. 1.1.

1.2.2 Examples

The practical relevance of CHPs is illustrated by the following two examples.

Example 1 (Trading with Macroeconomic News) Rambaldi et al. (2015) analyze the impact of macroeconomic news on market activity, measured by the rate of change in the best quotes. They consider a Hawkes process driven by an endogenous and an exogenous kernel. The self-excitation effect is described by an unmarked endogenous kernel in the form of a linear combination of exponentials,

$$g(t) = \alpha_A e^{-\beta_A t} + \alpha_B e^{-\beta_B t},$$

and the effect of (recurring) macroeconomic news by an exogenous kernel in the form of a single exponential,

$$\Phi_N(t) = \alpha_N e^{-\beta_N t}.$$

While best-quote changes occur at the random instants τ_i, the arrival times ϑ_j of news releases are known in advance, rendering the exogenous kernel deterministic. Provided the control function $a(t) \equiv \sum_{j:\vartheta_j<t} \Phi_N(t-\vartheta_j)$, the intensity of the controlled Hawkes process is then given by Eq. (1.1).

Example 2 (Credit Collections) A CHP was used by Chehrazi and Weber (2015) to predict the repayment behavior by holders of credit-card accounts in default. A delinquent account with outstanding balance $B(0) > 0$ placed into collections at time $t = 0$ is credited with relative repayments r_i at times $\{\tau_i\}_{i\in\mathbb{N}}$ until the outstanding debt is paid in full. The sequence (τ_i, r_i),

for $i \geq 1$, constitutes a marked point process with intensity dynamics that can be described by a mean-reverting stochastic differential equation together with an initial condition:

$$\begin{aligned} d\lambda(t) &= \underbrace{\kappa\left(\lambda_\infty - \lambda(t)\right) dt}_{\text{mean-reversion}} + \underbrace{\delta_1^\top dJ(t)}_{\text{self-excitation}} + \underbrace{da(t)}_{\text{control}}, \\ \lambda(0) &= \lambda_0, \end{aligned} \tag{1.3}$$

where the two-dimensional jump process $J(t) = [N(t), R(t)]^\top$ represents the marked and unmarked version of the same counting process; indeed, $N(t) = \sum_i \mathbb{1}_{\{\tau_i < t\}}$ captures the holder's willingness to pay and $R(t) = \sum_i \mathbb{1}_{\{\tau_i < t\}} r_i$ his or her ability to pay. The parameter λ_∞ represents the long-run steady-state to which the repayment intensity reverts at the rate κ, while $\delta_1 = [\delta_{10}, \delta_{11}]$ denotes the sensitivity to the self-exciting two-dimensional jumps. The control variable $a(t)$ is assumed to be a deterministic nondecreasing right-continuous and piecewise-constant function, taking values in $\mathbb{R}_+$. The exogenous kernel is

$$a(t; \delta_2, \kappa) = \sum_{j:\vartheta_j < t} \delta_{2l(\vartheta_j)} e^{-\kappa(t-\vartheta_j)},$$

where the parameter vector $\delta_2 = (\delta_{21}, \delta_{22}, \ldots, \delta_{2M})$ contains the sensitivities of the repayment intensity to M different account-treatment actions, and the mapping $l : \mathbb{R}_+ \to \mathbb{N}_+$ describes the type $l(\vartheta_j)$ of the action taken at time ϑ_j. In practice, the impulses map to the available collector actions which can vary from mild (inducing smaller intensity jumps, e.g., by sending a letter of notice or making phone calls) to severe (inducing larger intensity jumps, e.g., by filing a lawsuit). Overall, the repayment intensity evolves according to Eq. (1.1) for $m_i \equiv r_i$. In this, the deterministic drift,

$$\mu(t) = \lambda_\infty + (\lambda_0 - \lambda_\infty) e^{-\kappa t},$$

is exponentially mean-reverting, and the triggering kernel,

$$g(t - \tau_i, r_i) = (\delta_{10} + \delta_{11} r_i)\, e^{-\kappa(t-\tau_i)},$$

describes the effect of a repayment-event arrival at time τ_i on the repayment intensity for all $t \geq \tau_i$.

1.2.3 Branching Structure

The branching structure presents an augmented view of a Hawkes point process, consisting of the Poisson cluster-process representation introduced by Hawkes and Oakes (1974). It maps event arrivals to clusters, each of which begins with an immigrant arrival following an inhomogeneous Poisson process of base-rate intensity $\mu(t) + a(t)$. Subsequently, every immigrant generates its own offsprings following an inhomogeneous Poisson process with intensity given by the triggering kernel $g(t)$, and this cascades through all offsprings, thus generically clustering the event arrivals. Conceptually, all events fall into two categories: immigrant arrivals and offspring arrivals. Offspring events are triggered by existing events in

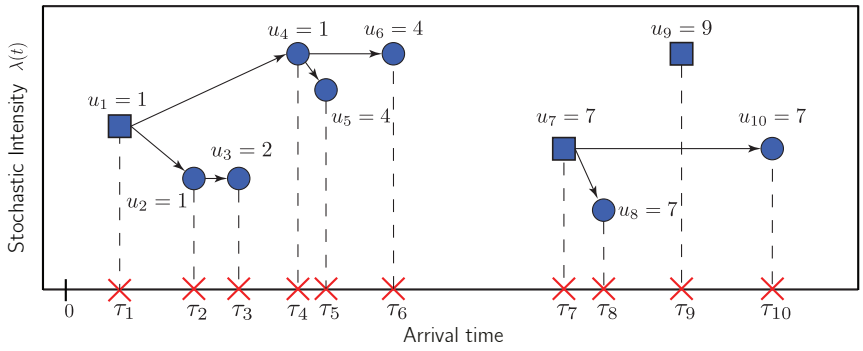


Figure 1.2: Branching-structure representation with immigrants (■) and offsprings (●) at the arrival times τ_i (×).

the process, while immigrant events arrive independently without being preceded by a parent event. This conceptual separation provides additional inner structure to the process.[4] More specifically, the branching structure of the i-th arrival at time τ_i is described by a mapping $u : \mathbb{R}_+ \to \mathbb{N}_+$, so

$$u_i = u(\tau_i), \quad i \geq 1, \tag{1.4}$$

where

$$u_i = \begin{cases} i, & \text{if arrival } i \text{ is an immigrant arrival,} \\ j, & \text{if the immediate ancestor of arrival } i \text{ is arrival } j. \end{cases}$$

Assigning either the immediate ancestor $j < i$ (if it exists), or else the current event i, the variable $u_i \in \{1, \dots, i\}$ determines the branching structure of the Hawkes process, by means of the marked point process (τ_i, u_i); see Fig. 1.2.

In practice, the branching structure is usually unobservable. Yet, conditional on a set of process parameters and a sample sequence (τ_i, m_i), it is possible to recover its probability distribution,

$$\mathbb{P}[u_i = i | \mathcal{H}_t] = \frac{\mu(\tau_i) + a(\tau_i)}{\lambda(\tau_i)} \qquad \text{and} \qquad \mathbb{P}\big[u_i = j | \mathcal{H}_t\big] = \frac{g(\tau_i - \tau_j, m_i)}{\lambda(\tau_i)}, \quad 1 \leq j < i. \tag{1.5}$$

Thus, it is possible to probabilistically assign any arrival i to being an immigrant or offspring (Fig. 1.3b).

As shown in Fig. 1.3a, a Hawkes process can be decomposed into a base-rate process and a sum of arrival-triggered inhomogeneous Poisson processes. The resulting (probabilistic) branching structure can be used to perform efficient numerical simulation (Møller and Rasmussen, 2006),

[4] See Daley and Vere-Jones (2003) for details on the theory of branching processes.

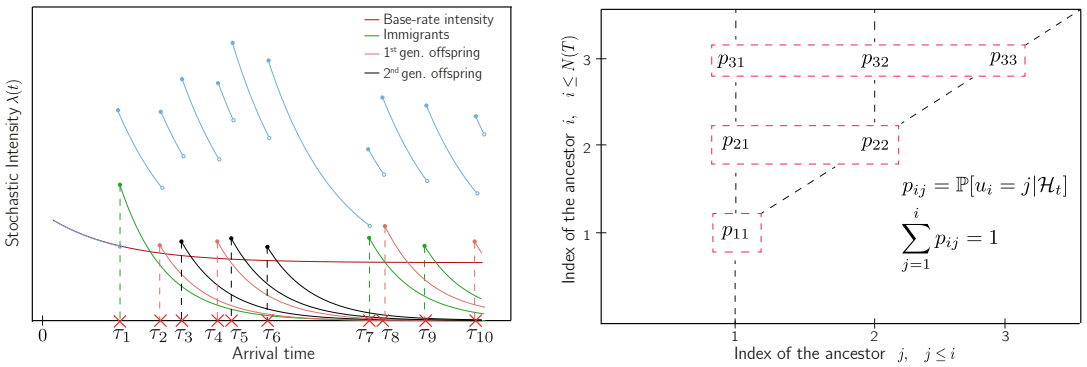


Figure 1.3: a) Intensity decomposition of the Hawkes process from Fig. 1.2 into a base-rate process and arrival-triggered inhomogeneous Poisson processes. Notice that the ratio of the intensity induced due to a particular arrival to the total intensity determines the probability that the event is an offspring of that particular arrival. b) Representation of the branching distribution: each row designates the probability mass function for the particular arrival; the diagonal represents probabilities of events being immigrants.

or as we show in Section 1.3, to improve process identification.

Remark 1 The branching structure implies an intuitive iso-perimetric constraint on the triggering kernel g that ensures the stability of the system. Indeed, the average number of direct (i.e., first-order) offsprings generated by a single event is the expected branching ratio $\nu = \mathbb{E}_m\left[\int_0^\infty g(t,m)\,dt\right]$, whereby the point process remains stable if and only if $\nu < 1$.[5]

1.3 Identification

Our estimation procedure is presented in the context of Ex. 2 concerning the collection on defaulted credit-card accounts. Repayments follow a CHP with intensity described by Eq. (1.1), conditional on the parameter vector $\theta = (\kappa, \lambda_0, \lambda_\infty, \delta_{10}, \delta_{11}, \delta_2)$, a known distribution F of the relative repayments (marks) $m_i = r_i$, and a given sequence of account-treatment times $\{\vartheta_j\}_{j\in\mathbb{N}}$. The information from a realization of such a process then consists of event times $\{\tau_i\}_{i\in\mathbb{N}}$, associated marks $\{r_i\}_{i\in\mathbb{N}}$ (representing a sample draw from the relative-repayment distribution F) and associated account-treatment times $\{\vartheta_j\}_{j\in\mathbb{N}}$. Note that the components $\delta_2^{(j)}$ of the parameter vector δ_2 usually take values in a finite set $\mathscr{D}$ with n_A elements, corresponding to the finitely many available actions, some of which (e.g., phone calls or text messages) may be applied repeatedly to the same account.

In the remainder of this section, we assume that K paths $\mathscr{H}_T^k$ (for $k \in \mathscr{K}$) have been observed

[5] Stability is defined as "nonexplosiveness" of the process in the sense that the ratio of total events $N(t)$ to the number $M(t) = \int_0^t (\mu(s) + a(s))\,ds$ of immigrant events remains bounded with probability 1. The stability criterion of $\nu < 1$ obtains, since for large t it is $N(t) \approx M(t)/(1-\nu)$, by the geometric-series formula.

over a finite time interval $[0,T]$, corresponding to an account portfolio $\mathcal{K} = \{1,\dots,K\}$. The joint information is summarized by $\mathcal{H}_T^{\mathcal{K}} = \{\mathcal{H}_T^k : k \in \mathcal{K}\}$.

1.3.1 Maximum-Likelihood Estimation

The conventional MLE-procedure directly solves

$$\begin{aligned} &\max_{\theta\in\Theta} && \ln \mathscr{L}\left(\theta|\mathcal{H}_T^{\mathcal{K}}\right), \\ &\text{subject to} && \theta \geq 0. \end{aligned} \tag{M}$$

where the *incomplete* data log-likelihood is given by

$$\ln \mathscr{L}(\theta|\mathcal{H}_T^{\mathcal{K}}) = \sum_{k=1}^{K} \left(-\int_0^T \lambda(s|\theta,\mathcal{H}_s^k)\,ds + \int_0^T \ln\lambda(s|\theta,\mathcal{H}_s^k)\,dN(s) \right). \tag{1.6}$$

The descriptor *incomplete* was coined by Veen and Schoenberg (2008); it emphasizes the fact that the estimator does not use additional branching-structure information. The incomplete log-likelihood estimator derived in this manner is asymptotically normal, efficient, and consistent. However, it suffers from the following two notable defects that significantly deteriorate its performance:

a) A closed-form solution to the maximization problem (M) is rarely available. Moreover, the efficiency of first- and second-order numerical methods is often poor, as in many cases the log-likelihood is extremely flat; see Fig. 1.4. Along certain trajectories even large disturbances reduce the log-likelihood only marginally. For instance, in the (λ_∞,κ)-subspace the parameter λ_∞ can be increased by a factor of 2 without significantly impacting the objective function.

b) Even in the simplest case of a constant-rate exponential Hawkes process, the log-likelihood can be multimodal (Ogata and Akaike, 1982). Specifically, the log-likelihood is concave only in the case where κ is fixed. For more complicated models, such as the case of the repayment process in Ex. 2, the optimization program is guaranteed to be nonconvex. Even if the log-likelihood is unimodal, due to the extreme flatness near the MLE-estimates $\hat{\theta}$, the objective function can become numerically multimodal as a result of rounding errors.

Although the main focus is to showcase how the branching structure can be employed in the estimation, we note several possible workarounds for MLE-convergence problems. The simplest solution to prevent the MLE-estimator from getting stuck at a local minimum is to solve the optimization program (M) in parallel for a large batch of starting values and then select the solution that achieves the highest log-likelihood. Although effective, the main drawback of this method is its computational cost, as will be shown in Section 1.4; the

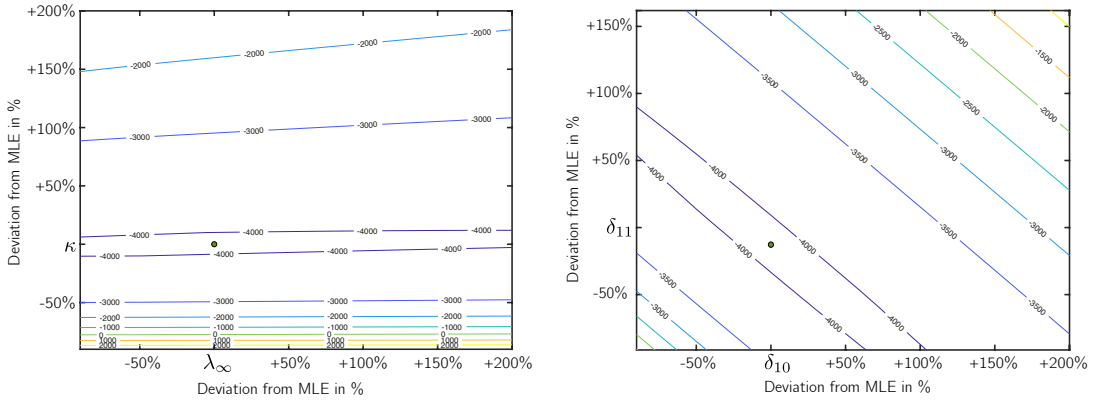


Figure 1.4: Flatness of the log-likelihood in multidimensional settings. The flatness is usually aggravated in a multidimensional context. Pairs of components of θ are varied around their MLEs $\hat{\theta}$ (green dot), while all other components remain fixed. a) Variation of κ. b) Variation of δ_{11}.

advantage of this method, compared to the EM-based algorithm presented below, is negligible. Another highly popular technique relies on the regularization of the estimator, imposing a coefficient penalty in an $\mathscr{L}_1$- or $\mathscr{L}_2$-norm (Zhou et al., 2013; Valera and Gomez-Rodriguez, 2015). In the context of Hawkes processes, Guo et al. (2018) proved that the regularized estimator is stable. However, the exact effects of the regularizer on the convergence are still not well understood; that is, despite being functional in practice, it does remain a "black-box solution."

1.3.2 Expectation-Maximization Algorithm

The expectation-maximization algorithm is based on the branching-structure representation introduced in Section 1.2.3. The idea is to provide the estimator with additional structural information about the process conditional on the observed sample in order to improve the fitting procedure, with the aim of circumventing the problems of ill-conditioning and lack of convergence that are prevalent in the standard MLE-procedure.

Complete Maximum-Likelihood Estimator. For a known branching structure described with the mapping u in Eq. (1.4), one obtains the *complete* data log-likelihood function as a sum of two terms, L_1 and L_2.

(i) Log-likelihood for immigrant events arriving with base-rate intensity $\lambda_b(t|\mathscr{H}_t^k) = \mu(t) + \sum_{j:\vartheta_j<t} \Phi_j(t-\vartheta_j)$, where $\mu(t) = \lambda_\infty + (\lambda_0 - \lambda_\infty)\, e^{-\kappa t}$ is the deterministic intensity of the inhomogeneous Poisson process for the immigrants and $\Phi_j(t-\vartheta_j) = \delta_{2l(\vartheta_j)} e^{-\kappa(t-\vartheta_j)}$ is

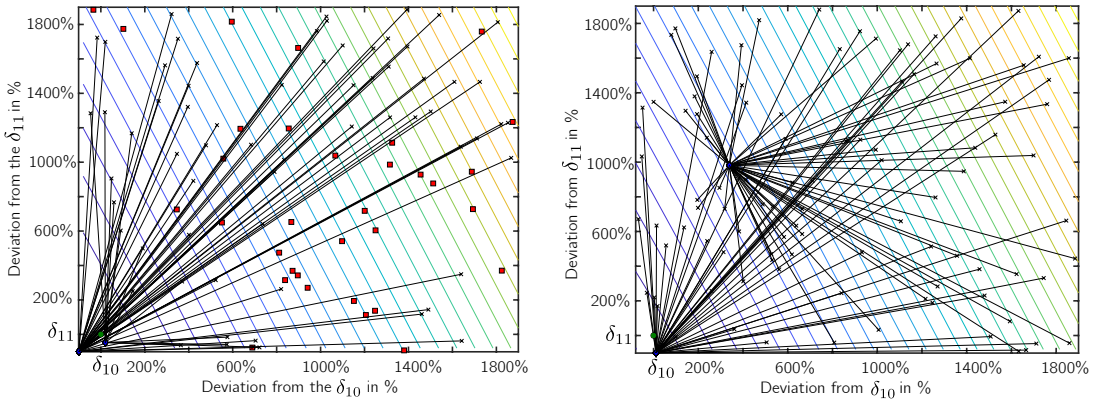


Figure 1.5: Convergence problems of the conventional MLE. Except for δ_{10} and δ_{11} the starting values for the estimation procedure are set to their reference values in θ_r; see Table 2.1. Black crosses denote starting values; blue diamonds denote estimation results; the green circle marks the location of the reference parameters. a) In 39 out of 100 cases, the optimization converged to absurdly large values and was registered as failed by red squares. b) Although the MLE-procedure converged in all 100 cases, the two discovered minima are local, both far from the reference parameters.

the effect of the action j carried out at time ϑ_j:

$$\begin{aligned} L_1\left(\kappa,\lambda_0,\lambda_\infty,\delta_2|\mathcal{H}_T^k,u\right) &= -\int_0^T \lambda_b(s|\mathcal{H}_s^k)\,ds + \int_0^T \ln\lambda_b(s|\mathcal{H}_s^k)\,dN(s) \\ &= -\left(\int_0^T \mu(s)\,ds + \sum_{j:\vartheta_j<T}\int_{\vartheta_j}^T \Phi_j(s-\vartheta_j)\,ds\right) \\ &\quad + \sum_{i:\tau_i\le T} \mathbb{1}_{\{u_i=i\}} \ln\lambda_b(\tau_i|\mathcal{H}_{\tau_i}^k) \\ &= -\left(\lambda_\infty T + \frac{1-e^{-\kappa T}}{\kappa}(\lambda_0-\lambda_\infty) + \sum_{j:\vartheta_j<T}\delta_{2\,l(\vartheta_j)}\frac{1-e^{-\kappa(T-\vartheta_j)}}{\kappa}\right) \\ &\quad + \sum_{i:\tau_i\le T} \mathbb{1}_{\{u_i=i\}} \ln\left(\mu(\tau_i) + \sum_{j:\vartheta_j<\tau_i}\delta_{2\,l(\vartheta_j)}e^{-\kappa(\tau_i-\vartheta_j)}\right), \end{aligned}$$

for all accounts $k\in\mathcal{K}$.

(ii) Cumulative log-likelihood of offspring events generated, respectively, by the different inhomogeneous Poisson processes with intensity $g(t-\tau_i,r_i)=(\delta_{10}+\delta_{11}r_i)\,e^{-\kappa(t-\tau_i)}$, for

$t \in [\tau_i, T]$:

$$L_2\left(\kappa, \delta_{10}, \delta_{11} | \mathcal{H}_T^k, u\right) = \sum_{i=1}^{N(T)} \left[-\int_{\tau_i}^{T} g(s-\tau_i, r_i)\, ds + \int_{\tau_i}^{T} \ln g(s-\tau_i, r_i)\, dN(s) \right]$$
$$= \sum_{i=1}^{N(T)} \left[-\int_{\tau_i}^{T} g(s-\tau_i, r_i)\, ds + \sum_{j=i+1}^{N(T)} \mathbb{1}_{\{u_j=i\}} \ln g(\tau_j - \tau_i, r_i) \right],$$

for all accounts $k \in \mathcal{K}$.

Summing $L_1 + L_2$ over the available sample paths in the account portfolio $\mathcal{K}$, the *complete* log-likelihood of the branching process, with intensity in Eq. (1.1), becomes

$$\ln \mathcal{L}_C\left(\theta | \mathcal{H}_T^{\mathcal{K}}, u\right) = \sum_{k=1}^{K} \left[L_1\left(\kappa, \lambda_0, \lambda_\infty, \delta_2 | \mathcal{H}_T^k, u\right) + L_2\left(\kappa, \delta_{10}, \delta_{11} | \mathcal{H}_T^k, u\right) \right]. \tag{C}$$

Note that the construction of the complete log-likelihood takes into account that the endogenous processes generating the offspring arrivals are mutually independent and independent of the exogenous process generating the immigrant arrivals (Hawkes and Oakes, 1974).

As the branching structure is unobservable, the complete log-likelihood is generally unavailable. It is therefore natural to resort to the *expected* complete log-likelihood (ECLL), conditional on the observed portfolio history $\mathcal{H}_T^{\mathcal{K}}$:

$$\mathbb{E}\left[\ln \mathcal{L}_C\left(\theta | \mathcal{H}_T^{\mathcal{K}}\right)\right] = \sum_{k=1}^{K} \mathbb{E}\Big[-\int_0^T \lambda_b(s|\theta, \mathcal{H}_s^k)\, ds + \sum_{i=1}^{N(T)} \mathbb{1}_{\{u_i=i\}} \ln \lambda_b(\tau_i|\theta, \mathcal{H}_s^k)$$
$$- \sum_{i=1}^{N(T)} \int_{\tau_i}^{T} g(s-\tau_i, r_i)\, ds + \sum_{i=2}^{N(T)} \sum_{j=1}^{i-1} \mathbb{1}_{\{u_i=j\}} \ln g(\tau_i - \tau_j, r_j) \Big].$$

Using the identity $\mathbb{E}\left[\mathbb{1}_{\{u_i=j\}} | \theta, \mathcal{H}_T^k\right] = \mathbb{P}\left[u_i = j | \theta, \mathcal{H}_T^k\right]$ together with Eq. (1.5), we obtain the ECLL:

$$\mathbb{E}\left[\mathcal{L}_C\left(\theta | \mathcal{H}_T^{\mathcal{K}}\right)\right] = \sum_{k=1}^{K} \Big[-\int_0^T \lambda_b(s|\theta, \mathcal{H}_T^k)\, ds + \sum_{i=1}^{N(T)} \mathbb{P}\left[u_i = i | \theta, \mathcal{H}_T^k\right] \ln \lambda_b(\tau_i|\theta, \mathcal{H}_T^k)$$
$$- \sum_{i=1}^{N(T)} \int_{\tau_i}^{T} g(s-\tau_i, r_i)\, ds + \sum_{i=2}^{N(T)} \sum_{j=1}^{i-1} \mathbb{P}\left[u_i = j | \theta, \mathcal{H}_T^k\right] \ln g(\tau_i - \tau_j, r_j) \Big]. \tag{EC}$$

EM-Algorithm. The expectation-maximization algorithm is initialized with a parameter value θ_0 obtained by using prior experience or an educated guess. The first step of the two-step iteration procedure (in iteration $n \geq 1$) consists of computing the *conditional* ECLL of the branching structure, termed $Q(\theta, \theta_n)$, by conditioning the probability distribution of the branching structure in Eq. (1.5) on the best available parameter estimate θ_n and the process parameters on the unknown parameter θ and the available portfolio data $\mathcal{H}_T^{\mathcal{K}}$. In the second step, one then performs a maximization of $Q(\theta, \theta_n)$ with respect to θ, resulting in the next

iterate: θ_{n+1}.

Expectation Step (E-Step). Using standard notation from the unsupervised-learning literature, where the EM-algorithm is frequently used for clustering purposes (John Lu 2010), the *conditional* ECLL becomes

$$Q(\theta,\theta_n) = \sum_{k=1}^{K}\Big[-\int_0^T \lambda_b(s|\theta,\mathscr{H}_T^k)\,ds + \sum_{i=1}^{N(T)} \mathbb{P}\Big[u_i = i|\theta_n,\mathscr{H}_T^k\Big]\ln\lambda_b(\tau_i|\theta,\mathscr{H}_T^k) \\ - \sum_{i=1}^{N(T)}\int_{\tau_i}^T g(s-\tau_i,r_i)\,ds + \sum_{i=2}^{N(T)}\sum_{j=1}^{i-1}\mathbb{P}\Big[u_i = j|\theta_n,\mathscr{H}_T^k\Big]\ln g(\tau_i-\tau_j,r_j)\Big]. \tag{1.7}$$

Note that the endogenous kernel g is computed conditional on the "true" parameter θ.

Maximization Step (M-Step). Based on the current parameter estimate θ_n the next iterate is determined as a result of maximizing the conditional ECLL:

$$\theta_{n+1} \in \arg\max_{\theta\in\Theta} Q(\theta,\theta_n), \tag{1.8}$$

where the compact parameter set Θ is a subset of the positive orthant, chosen by the user so as to limit the search using standard numerical tools.

Termination. Starting with the initial seed θ_0, one iterates through the Expectation and Maximization steps until the termination condition,

$$Q(\theta_{n+1},\theta_{n+1}) - Q(\theta_n,\theta_n) \le \varepsilon, \tag{1.9}$$

is satisfied for a sufficiently small tolerance $\varepsilon > 0$. The procedure is summarized hereafter.

Initialize seed $\theta_0 \in \Theta$, fix a tolerance $\varepsilon \in (0,1)$, and set $n \leftarrow 0$, $\delta \leftarrow 1$;
while $\delta > \varepsilon$ **do**
  E-Step: Calculate $\mathbb{P}\left[u_i = j|\theta_n,\mathscr{H}_T^{\mathcal{K}}\right]$ for all $1 \le j \le i \le N(T)$;
  M-Step: Find $\theta_{n+1} \in \arg\max_{\theta\in\Theta} Q(\theta,\theta_n)$;
  $n \leftarrow n+1$ and $\delta \leftarrow Q(\theta_{n+1},\theta_{n+1}) - Q(\theta_n,\theta_n)$;
end

Convergence. Dempster et al. (1977) show that the sequence $(Q(\theta_n,\theta_n))_{n\in\mathbb{N}}$ is increasing and bounded, so that it must converge (Rudin et al., 1976, p. 55). However, there is no guarantee that the limit of the maximizing sequence (see, e.g., Gelfand and Fomin, 1963, Ch. 8) is indeed associated with a global extremum. Conceptually, the EM-estimates are expected MLE-estimates. Dempster et al. (1977) also establish that estimates obtained using the EM-algorithm are consistent, just as standard MLE-estimates (based on the incomplete log-likelihood function).

Solving the MLE-problem (M) numerically usually entails local approximations of the objective

function, followed by choosing an appropriate increment in the direction of steepest ascent. By contrast, the EM-algorithm produces a local approximation of the objective conditional on the model parameters and the distribution of the branching structure as a latent variable. This local approximation constitutes a lower bound for the incomplete log-likelihood (Minka, 1998). The EM-algorithm alternates between updating the lower bound (E-step) in Eq. (1.7) and updating the parameter estimate (M-step) in Eq. (1.8) until the termination condition in Eq. (1.9) is satisfied. Thus, by construction, $\mathscr{L}(\hat{\theta}|\mathscr{H}_T^{\mathscr{K}}) \geq Q(\hat{\theta},\hat{\theta})$, as shown in the remark below. Intuitively, direct maximization can be viewed as fitting a single point process with specified intensity function, whereas maximizing the conditional ECLL (via the EM-algorithm) *simultaneously* fits $N(T)+1$ inhomogeneous Poisson processes,[6] each weighted by its corresponding branching-structure probability.

Remark 2 (EM produces lower bound for MLE) For simplicity we assume that $a = 0$ (or else consider $\hat{\mu} = \mu + a$ instead of μ). Comparing the classical incomplete log-likelihood and the expected complete log-likelihood (as in a log-likelihood-ratio test) yields

$$\begin{aligned}
\ln \mathscr{L} - \mathbb{E}[\ln \mathscr{L}_C] &= \sum_{i=1}^{N(T)} \ln \lambda(\tau_i) - \sum_{i=1}^{N(T)} \frac{\mu(\tau_i)}{\lambda(\tau_i)} \ln \mu(\tau_i) - \sum_{i=2}^{N(T)} \sum_{j=1}^{i-1} \frac{g(\tau_i - \tau_j, m_j)}{\lambda(\tau_i)} \ln g(\tau_i - \tau_j, m_j) \\
&= \sum_{i=2}^{N(T)} \Bigg[\ln \Bigg(\mu(\tau_i) + \sum_{j=1}^{i-1} g(\tau_i - \tau_j, m_j) \Bigg) - \frac{\mu(\tau_i)}{\lambda(\tau_i)} \ln \mu(\tau_i) \\
&\quad - \sum_{j=1}^{i-1} \frac{g(\tau_i - \tau_j, m_j)}{\lambda(\tau_i)} \ln g(\tau_i - \tau_j, m_j) \Bigg] \\
&= \sum_{i=2}^{N(T)} \Bigg[\ln \Bigg(\mu(\tau_i) + \sum_{j=1}^{i-1} g(\tau_i - \tau_j, m_j) \Bigg) \\
&\quad - \ln \Bigg(\mu(\tau_i)^{\mathbb{P}[u_i = i]} \prod_{j=1}^{i-1} g(\tau_i - \tau_j, m_j)^{\mathbb{P}[u_i = j]} \Bigg) \Bigg] \\
&= \sum_{i=2}^{N(T)} \ln \Bigg(\frac{\mu(\tau_i) + \sum_{j=1}^{i-1} g(\tau_i - \tau_j, m_j)}{\mu(\tau_i)^{\mathbb{P}[u_i = i]} \prod_{j=1}^{i-1} g(\tau_i - \tau_j, m_j)^{\mathbb{P}[u_i = j]}} \Bigg) \geq 0.
\end{aligned}$$

To obtain the last inequality, note first that μ and g have nonnegative values, and $\mathbb{P}[u_i = i] + \sum_{j=1}^{i-1} \mathbb{P}[u_i = j] = 1$, for all $i \in \{1, \ldots, N(T)\}$. Furthermore, it is

$$\mu(\tau_i) + \sum_{j=1}^{i-1} g(\tau_i - \tau_j) \geq \mathbb{P}[u_i = i]\mu(\tau_i) + \sum_{j=1}^{i-1} g(\tau_i - \tau_j, m_j)\mathbb{P}[u_i = j].$$

By the concavity of the natural logarithm and Jensen's inequality we get

$$\ln \Bigg(\mathbb{P}[u_i = i]\mu(\tau_i) + \sum_{j=1}^{i-1} g(\tau_i - \tau_j, m_j)\mathbb{P}[u_i = j] \Bigg) \geq \mathbb{P}[u_i = i] \ln \mu(\tau_i) + \sum_{j=1}^{i-1} \mathbb{P}[u_i = j] \ln g(\tau_i - \tau_j, m_j).$$

[6] Any immigrant arrival triggers an offspring process and so does each offspring arrival. Hence, there is a process for each arrival (altogether $N(T)$ processes) and one immigrant process. The $N(T)+1$ processes are coupled by the branching distribution in Eq. (1.5), which depends on the parameter vector and the observed sample data.

The right-hand side can then be rewritten in the form

$$\ln \mu(\tau_i)^{\mathbb{P}[u_i=j]} + \sum_{j=1}^{i-1} \ln g(\tau_i - \tau_j, m_j)^{\mathbb{P}[u_i=j]} = \ln\left(\mu(\tau_i)^{\mathbb{P}[u_i=i]} \prod_{j=1}^{i-1} g(\tau_i - \tau_j, m_j)^{\mathbb{P}[u_i=j]}\right).$$

Finally, given that the logarithm is an increasing function, it is

$$\mu(\tau_i) + \sum_{j=1}^{i-1} g(\tau_i - \tau_j, m_j) \geq \mu(\tau_i)^{\mathbb{P}[u_i=j]} \prod_{j=1}^{i-1} g(\tau_i - \tau_j, m_j)^{\mathbb{P}[u_i=j]},$$

which implies the inequality in question.

Although the objective function in Eq. (1.7) minorizes the log-likelihood, we note that this lower bound is generally not tight. By taking into account the entropy of the branching distribution, the following characterization result corrects this shortcoming and provides a tight lower bound, guaranteeing that the approximation of the log-likelihood becomes exact at the optimal EM-estimate.

Theorem 1 (Representation) *For all $\theta \in \Theta$, the incomplete log-likelihood can be written in the form*

$$\ln \mathscr{L}(\theta|\mathscr{H}_T^K) = Q(\theta,\theta) + \Delta(\theta), \tag{1.10}$$

where the nonnegative defect,

$$\Delta(\theta) = -\sum_{k=1}^{K} \sum_{i=1}^{N(T)} \sum_{j=1}^{i} \mathbb{P}[u_i = j|\theta, \mathscr{H}_T] \ln \mathbb{P}[u_i = j|\theta, \mathscr{H}_T], \quad (\geq 0) \tag{1.11}$$

describes the entropy of the branching distribution given the observed history $\mathscr{H}_T$.

Proof of Thm. 1. Without any loss of generality, we set $K = 1$, so that there is only a single data path in a singleton portfolio, with the superscript k dropped for notational convenience. Assume a realization $\boldsymbol{X} = \{(\tau_1, r_1), (\tau_2, r_2), \ldots, (\tau_n, r_n)\}$ of a CHP given by Eq. (1.1) with a branching structure described with a latent variable $\boldsymbol{Y} = \{y_1, y_2, \cdots, y_n\}$ (i.e., y_i denotes the ancestor of the i-th arrival).[7] That is, $\boldsymbol{X}$ is the incomplete data with complete data given by $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})$. Furthermore, we assume a density of the observed variable $p(\boldsymbol{X}|\theta)$, an arbitrary density of the latent variable $q(\boldsymbol{Y})$, and a joint density $p(\boldsymbol{X}, \boldsymbol{Y}|\theta)$ between the observed and hidden variables. In the setting of CHPs, we can identify the first and the last of the densities with the incomplete and complete log-likelihoods, i.e.,

$$p(\boldsymbol{X}|\theta) = \mathscr{L}(\theta|\boldsymbol{X}), \tag{1.12}$$

$$p(\boldsymbol{X}, \boldsymbol{Y}|\theta) = \mathscr{L}_C(\theta|\boldsymbol{X}, \boldsymbol{Y}). \tag{1.13}$$

[7] The index n describes the total number of arrivals, i.e., $n = N(T)$.

Let G be a lower bound to the log-likelihood function parametrized by a parameter θ and the density $q(\boldsymbol{Y})$, such that

$$G(\theta, q) = \ln \mathscr{L}(\theta|\boldsymbol{X}) - D\big(q \parallel p(\cdot|\boldsymbol{X},\theta)\big) \leq \ln \mathscr{L}(\theta|\boldsymbol{X}), \tag{1.14}$$

where $D\big(q \parallel p(\cdot|\boldsymbol{X},\theta)\big)$ denotes the Kullback-Leibler divergence (relative entropy) of q with respect to $p(\cdot|\boldsymbol{X},\theta)$.[8] Clearly, the bound G becomes tight if and only if the two distributions are identical. The tight lower bound G can therefore be expressed (for $q(\cdot) = p(\cdot|X,\theta)$) as

$$\begin{aligned} G(\theta, q) &= \ln p(\boldsymbol{X}|\theta) - \mathbb{E}_{\boldsymbol{Y}}\left[\ln \frac{p(\boldsymbol{Y}|\boldsymbol{X},\theta)}{p(\boldsymbol{Y}|\boldsymbol{X},\theta)}\right] = \mathbb{E}_{\boldsymbol{Y}}\left[\ln \frac{p(\boldsymbol{Y}|\boldsymbol{X},\theta)}{p(\boldsymbol{Y}|\boldsymbol{X},\theta)} + \ln p(\boldsymbol{X}|\theta)\right] \\ &= \mathbb{E}_{\boldsymbol{Y}}\left[\ln \frac{p(\boldsymbol{Y}|\boldsymbol{X},\theta)\,p(\boldsymbol{X}|\theta)}{p(\boldsymbol{Y}|\boldsymbol{X},\theta)}\right] = \mathbb{E}_{\boldsymbol{Y}}\left[\ln \frac{p(\boldsymbol{X},\boldsymbol{Y}|\theta)}{p(\boldsymbol{Y}|\boldsymbol{X},\theta)}\right] \\ &= \mathbb{E}_{\boldsymbol{Y}}\left[\ln p(\boldsymbol{X},\boldsymbol{Y}|\theta)\right] - \mathbb{E}_{\boldsymbol{Y}}\left[\ln p(\boldsymbol{Y}|\boldsymbol{X},\theta)\right], \end{aligned}$$

where the penultimate equality is obtained using the law of total probability. The two terms correspond to the ECLL in Eq. (1.7) and the adjustment term $\Delta(\theta)$ in Eq. (1.11), respectively. Consequently, the branching distribution $p(\cdot|\boldsymbol{X},\theta)$ is given by

$$p(\boldsymbol{Y}|\boldsymbol{X},\theta) = \prod_{i=1}^{n} \mathbb{P}[u_i = y_i|\theta,\boldsymbol{X}],$$

for any branching-structure realization $\boldsymbol{Y}$ (see also Fig. 1.3). Finally, we recover

$$\begin{aligned} \Delta(\theta) &= -\mathbb{E}_{\boldsymbol{Y}}\left[\ln p(\boldsymbol{Y}|\theta,\boldsymbol{X})\right] = -\mathbb{E}_{\boldsymbol{Y}}\left[\ln \prod_{i=1}^{n} \mathbb{P}[u_i = y_i|\theta,\boldsymbol{X}]\right] = -\mathbb{E}_{\boldsymbol{Y}}\left[\sum_{i=1}^{n} \ln \mathbb{P}[u_i = y_i|\theta,\boldsymbol{X}]\right] \\ &= -\sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{Y}}\left[\ln \mathbb{P}[u_i = y_i|\theta,\boldsymbol{X}]\right] = -\sum_{i=1}^{n}\sum_{j=1}^{i} \mathbb{P}[u_i = j|\boldsymbol{X},\theta] \ln \mathbb{P}[u_i = j|\theta,\boldsymbol{X}], \end{aligned}$$

which concludes the proof. ■

Thm. 1 implies that the conditional ECLL $Q(\theta,\theta_n)$ can be "adjusted" using the defect Δ to become a tight lower bound for the log-likelihood, as follows:

$$\hat{Q}(\theta,\theta_n) = Q(\theta,\theta_n) + \Delta(\theta). \tag{1.15}$$

This *adjusted* (conditional) ECLL can be written in the form

$$\begin{aligned} \hat{Q}(\theta,\theta_n) = \sum_{k=1}^{K} \Big[&-\int_0^T \lambda_b(s|\theta,\mathscr{H}_T^k)\,ds + \sum_{i=1}^{N(T)} \mathbb{P}\left[u_i = i|\theta_n,\mathscr{H}_T^k\right] \ln \frac{\lambda_b(\tau_i|\theta,\mathscr{H}_T^k)}{\mathbb{P}\left[u_i = i|\theta_n,\mathscr{H}_T^k\right]} \\ &- \sum_{i=1}^{N(T)} \int_{\tau_i}^{T} g(s-\tau_i, r_i)\,ds + \sum_{i=2}^{N(T)}\sum_{j=1}^{i-1} \mathbb{P}\left[u_i = j|\theta_n,\mathscr{H}_T^k\right] \ln \frac{g(\tau_i - \tau_j, r_j)}{\mathbb{P}\left[u_i = j|\theta_n,\mathscr{H}_T^k\right]} \Big]. \end{aligned}$$

[8] While not being a proper metric, the Kullback-Leibler divergence is nonnegative (Gibbs' inequality), and it vanishes if and only if the two distributions in its argument coincide almost everywhere.

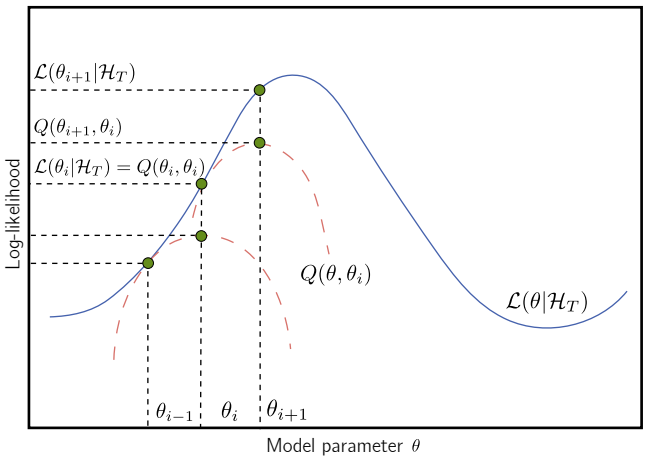


Figure 1.6: Illustration of a single EM-iteration. Each E-step calculates the branching distribution and determines a functional form of the lower bound that is then maximized in the M-step.

The adjusted ECLL not only establishes direct comparability with the incomplete log-likelihood, but it also significantly reduces the number of iterations needed for convergence. Henceforth, all mentions of "ECLL" refer to the *adjusted* ECLL with objective function $\hat{Q}$ (instead of Q).

Building on the proof of Thm. 1 (preserving the notation used there), the EM-algorithm can be described as follows.

Initialize seed $\theta_0 \in \Theta$, fix a tolerance $\varepsilon \in (0,1)$, and set $n \leftarrow 0$, $\delta \leftarrow 1$;
while $\delta > \varepsilon$ **do**
  E-Step: Calculate $q_{n+1} \in \arg\max_q G(\theta_n, q) = \{p(\boldsymbol{Y}|\boldsymbol{X},\theta_n)\}$;
  M-Step: Find $\theta_{n+1} \in \arg\max_{\theta\in\Theta} G(\theta, q_{n+1})$;
  $n \leftarrow n+1$ and $\delta \leftarrow G(\theta_{n+1}, q_{n+1}) - G(\theta_n, q_n)$;
end

The E-Step determines the next density q_{n+1} of the latent variable $\boldsymbol{Y}$ (i.e., the branching distribution), based on the current parameter estimate θ_n, by maximizing the adjusted (conditional) ECLL $G(\theta_n,\cdot)$; by Eq. (1.14) the maximum is equal to the incomplete log-likelihood and is achieved at $q_{n+1} = p(\cdot|\boldsymbol{X},\theta_n)$. The M-Step then provides the next parameter estimate θ_{n+1} by maximizing the adjusted ECLL $G(\cdot, q_{n+1})$ on the compact set Θ. Visually, Alg. 2 is represented in Fig. 1.6.

1.4 Simulation

For a systematic comparison of the proposed EM-algorithm with the standard MLE-procedure, we are particularly interested in its convergence performance with respect to randomized

κ	λ_0	λ_∞	δ_{10}	δ_{11}	$\delta_2^{(1)}$	$\delta_2^{(2)}$	$\delta_2^{(3)}$
0.4	0.05	0.03	0.08	0.06	0.03	0.06	0.09

Table 1.1: Specification of the reference repayment process (θ_r) for the numerical experiment.

initial values θ_0. Convergence performance is key in practice, since an appropriate parameter range is difficult to determine *ex ante*. Even "educated guesses" for θ_0 are bound to often stray significantly from the ("true") reference value θ_r. The latter is used in our broad numerical experiment to generate synthetic collections data in the context of Ex. 2.

1.4.1 Data

It is important to note that credit-collections data by their very nature are relatively sparse. A significant portion of accounts does not exhibit any repayments.[9] This is compensated by the transversal experience across an account portfolio $\mathscr{K}$ containing K sample paths. Throughout the numerical experiment, we consider a CHP driven by the intensity in Eq. (1.1), generated with the reference parameters specified in Table 2.1. The marks (relative repayments) are assumed to be independent and identically distributed (i.i.d.), uniformly (i.e., $r_i \sim U[0,1]$).

For each of $K = 500$ accounts in the portfolio $\mathscr{K}$, we consider $L = 100$ sample paths, referred to as "scenarios." Each scenario $\ell \in \{1,\dots,L\}$ generates a history $\mathscr{H}_T^{\mathscr{K}}(\ell)$, based on which the model identification is performed using the two alternative methods (MLE and EM). This is done for $M = 100$ random seeds $\theta_0^{(1)},\dots,\theta_0^{(M)}$, obtained as realizations of the random variable $\theta_r \operatorname{diag}(\gamma)$, where $\gamma = (\gamma_g)$ is a vector of the same length as θ_r with entries of the form $\gamma_g = 10^{\beta_g/(20\,\mathrm{dB})}$ describing the gain (positive or negative). In the numerical experiment, gains are considered to be such that $\beta_g \in [-26\,\mathrm{dB}, 26\,\mathrm{dB}]$, corresponding to amplitude distortions γ_g in the interval $[1/20, 20]$.[10]

Each scenario history $\mathscr{H}_T^{\mathscr{K}}(\ell)$ corresponds to data from a portfolio of K treated accounts, with observation horizon $T = 100$, where each account $k \in \mathscr{K}$ is associated with an observed repayment sequence $\{(\tau_i^k, r_i^k)\}$ and a sequence of three control impulses (account treatments) at the i.i.d. times $\vartheta_1^k \sim U[0,T]$ (chosen such that $\vartheta_1^k < \vartheta_2^k < \vartheta_3^k$).

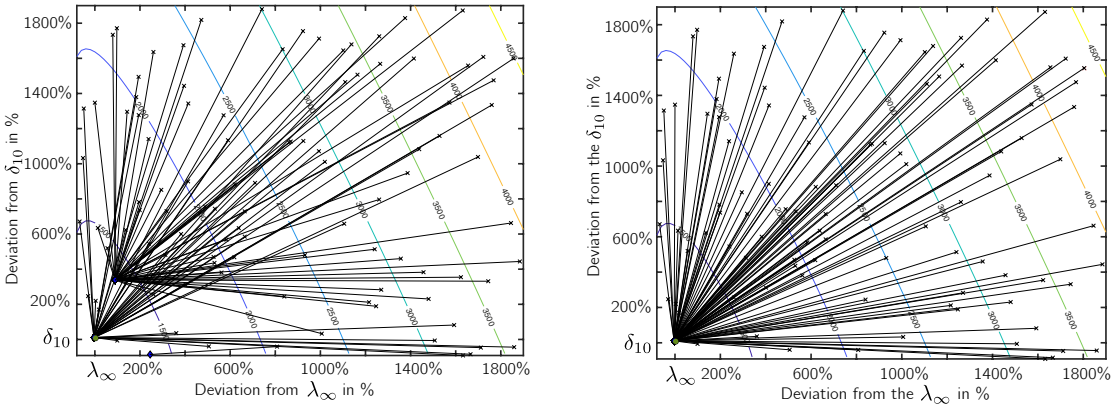
Table 1.2 compares the average performance of the MLE-estimator $\hat{\theta}_{\mathrm{MLE}}$ and the EM-estimator $\hat{\theta}_{\mathrm{EM}}$ over 100 random seeds, distributed uniformly within ±25% of the reference parameter values. It also indicates how the length of the observation horizon T impacts the respective accuracy of the two estimators. Interestingly, both methods produce very similar estimates, although EM tends to be computationally more expensive. This behavior is somewhat expected, as both methods produce MLE-estimates with the EM-algorithm relying on additional

[9] Even an "empty" sample path conveys valuable information about the underlying process and thus cannot be discarded.

[10] Here we consider a uniform distribution (i.i.d.) of γ_g on $[1/20, 20]$. We have also run the entire study for a uniform distribution in the dB-space, i.e., for β_g uniformly distributed (i.i.d.) on the interval $[-26\,\mathrm{dB}, 26\,\mathrm{dB}]$, with similar results.

T		$\hat{\kappa}$	$\hat{\lambda}_0$	$\hat{\lambda}_\infty$	$\hat{\delta}_{10}$	$\hat{\delta}_{11}$	$\hat{\delta}_2^{(1)}$	$\hat{\delta}_2^{(2)}$	$\hat{\delta}_2^{(3)}$	Runtime	Mean $[N(T)]$
500	$\hat{\theta}_{\text{MLE}}$	0.3920	0.0457	0.0300	0.0823	0.0509	0.0307	0.0594	0.0982	90 s	23
	$\hat{\theta}_{\text{EM}}$									1,456 s	
	Bias	(-2.05%)	(-8.58%)	(+0.03%)	(+2.83%)	(-15.22%)	(+2.38%)	(-1.10%)	(+9.11%)		
1000	$\hat{\theta}_{\text{MLE}}$	0.3997	0.0473	0.0298	0.0777	0.0645	0.0332	0.0588	0.0905	155 s	42
	$\hat{\theta}_{\text{EM}}$									2,579 s	
	Bias	(-0.03%)	(-5.35%)	(-0.57%)	(-2.84%)	(+ 7.53%)	(+10.54%)	(-1.92%)	(+0.55%)		
2000	$\hat{\theta}_{\text{MLE}}$	0.4070	0.0471	0.0301	0.0817	0.0588	0.0359	0.0626	0.0968	426 s	85
	$\hat{\theta}_{\text{EM}}$									6,144 s	
	Bias	(+1.75%)	(-5.71%)	(+0.41%)	(+2.11%)	(-1.99%)	(+19.71%)	(+4.31%)	(+7.53%)		

Table 1.2: Asymptotic behavior of the MLE-estimator. Each estimate represents the average over 20 independently generated portfolios and 10 random starting values distributed ±25% around the respective (true) reference value.



a) MLE converging to three distinct points. b) EM converging close to the reference values.

Figure 1.7: Comparison of estimator convergence. Except for λ_∞ and δ_{10} the parameter starting values are set to their reference values in θ_r; see Table 2.1. The black crosses denotes starting values; blue diamonds denote estimation results; the red circle marks the location of the reference values λ_∞ and δ_{10}.

information related to the branching structure of the repayment process. The real advantage of the EM-algorithm over direct MLE-maximization becomes apparent when considering initial seeds θ_0 of significant distance to the reference parameter θ_r or when limiting the observation horizon.

1.4.2 Results

Consider first the convergence of the estimation in several two-dimensional subspaces of the parameter space, Θ. For this, the starting values $\theta_0^{(m)}$, for $m \in \{1,\dots,M\}$, are set to their reference value θ_r, while the investigated pair of parameter components are randomly varied between -26 dB and +26 dB (corresponding to a maximum variation by a factor of 20) relative to their corresponding reference values as starting values.

Fig. 1.7 shows that MLE fails to converge to the reference values for more than half of the initial

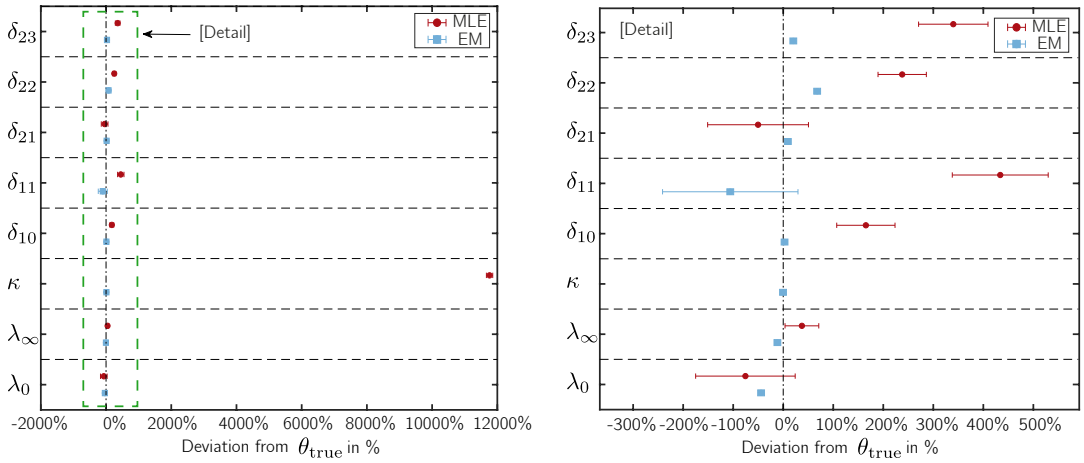


Figure 1.8: Estimation results for the scenario in Fig. 1.7. The center of the error plot designates the average bias over 100 starting values $\hat{\theta}^{(1)} \ldots \hat{\theta}^{(100)}$; error bands are based on one standard deviation. a) Full display of all parameters, with $\hat{\kappa}$ attaining high local maxima. b) Detail view near the zero-bias line.

seeds (in 52 of 100 instances). We note that the optimizer designated all of the estimation results (blue diamonds) as local minima (implying that a step in any direction would not improve the objective function). To reveal the performance of the estimator in the complete parameter space we use error plots investigating the relationship $\theta_0^{(m)} \rightarrow \hat{\theta}^{(m)}$. Fig. 1.8 indicates that while the EM-algorithm succeeds in bypassing erroneous local minima, it does so with reduced variance in the estimation results.

Another encountered deficiency is that the MLE-estimator failed to converge entirely for a subset of starting values, as can be seen in Fig. 1.5.[11] Again, this behavior was not registered for the EM-algorithm, except for cases with starting points deviating by more than +10,000% (corresponding to about 32 dB) from the reference values.

Remark 3 (Numerical Conditioning) The superior convergence performance of the EM-algorithm has two possible sources. First, the properties derived for the MLE-estimator hold only asymptotically. Although both $\hat{\theta}_{\text{MLE}}$ and $\hat{\theta}_{\text{EM}}$ are consistent, in a limited sample both estimates can differ, as

$$Q(\hat{\theta}_{\text{EM}}, \hat{\theta}_{\text{EM}}) \geq Q(\hat{\theta}_{\text{MLE}}, \hat{\theta}_{\text{MLE}})$$

and

$$\mathcal{L}(\hat{\theta}_{\text{MLE}}; \mathcal{H}_T^{\mathcal{K}}) \geq \mathcal{L}(\hat{\theta}_{\text{EM}}; \mathcal{H}_T^{\mathcal{K}}).$$

In our application, the number of repayments may not be large enough for attaining an asymptotic regime in the numerical maximization of the incomplete log-likelihood. On the

[11] An estimation run is counted as "failed" if any of the estimates exceeds a value of 10^3.

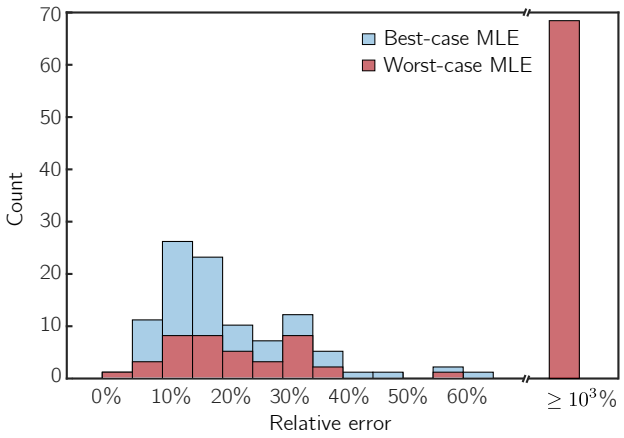


Figure 1.9: Sample distribution of the relative error for the best- and worst-case MLE-estimates, measured in terms of incomplete log-likelihood.

other hand, it might be enough for the EM-algorithm to produce accurate results, due to the additional branching-structure information captured by the EM-estimator.

Second, given the EM-construction as a lower bound for the MLE, intuitively, it is expected that the EM-objective function will exhibit a larger "curvature" compared to the incomplete log-likelihood. Indeed, as shown in Fig. 1.10, the objective function for the EM-algorithm appears to be a better conditioned objective function for the same problem. We showcase this property using the condition number of the Hessian matrix, which is intricately linked to the convergence performance. In particular, we focus on the difference between condition number for the MLE- and EM-surface; see Fig. 1.11. Computationally we obtain that the EM-objective shows a better conditioned Hessian on average in 80% of all points in the search space ($\theta_r \pm 32\,\mathrm{dB}$). Nevertheless, it is important to remember that both methods are local techniques, so neither can provide any guarantee for attaining a global maximizer.

Classical Benchmark. As indicated in Section 1.3.1 the erroneous local minima and hence the convergence issues can be circumvented using certain heuristic techniques. Disregarding for a moment the computational burden of repeating the optimization for M initial guesses, we characterize every batch of starting values by a single vector of estimates $\bar{\theta}$ that produces the largest incomplete log-likelihood for MLE and EM, respectively:

$$\bar{\theta}_{\mathrm{MLE}} \in \arg\max_{\hat{\theta}\in\hat{\Theta}_M} \mathscr{L}\big(\hat{\theta}|\mathscr{H}_T^{\mathscr{K}}\big),$$

and

$$\bar{\theta}_{\mathrm{EM}} \in \arg\max_{\hat{\theta}\in\hat{\Theta}_M} Q(\hat{\theta},\hat{\theta}),$$

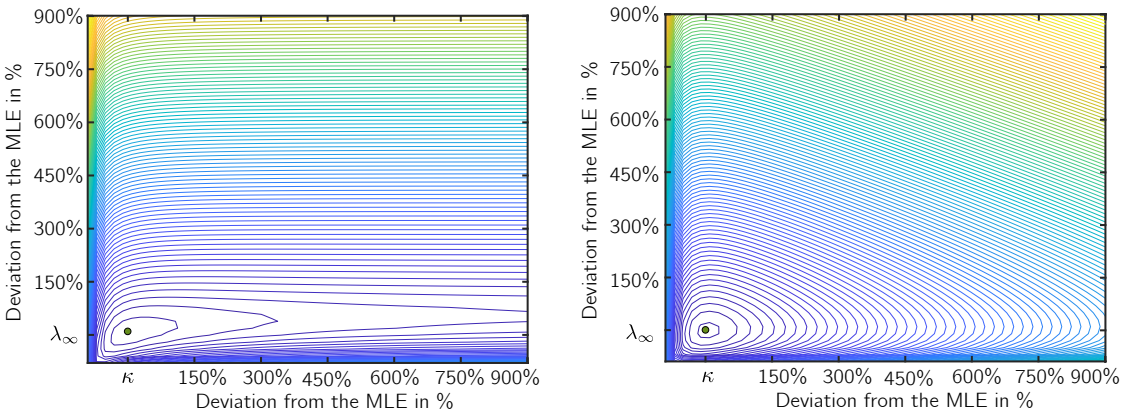


Figure 1.10: Comparison of the curvature of the objective functions. a) Incomplete log-likelihood exhibits a long valley around λ_∞. b) ECLL $Q(\theta,\theta_{n'})$, where n' is the last iterate before attaining the termination condition.

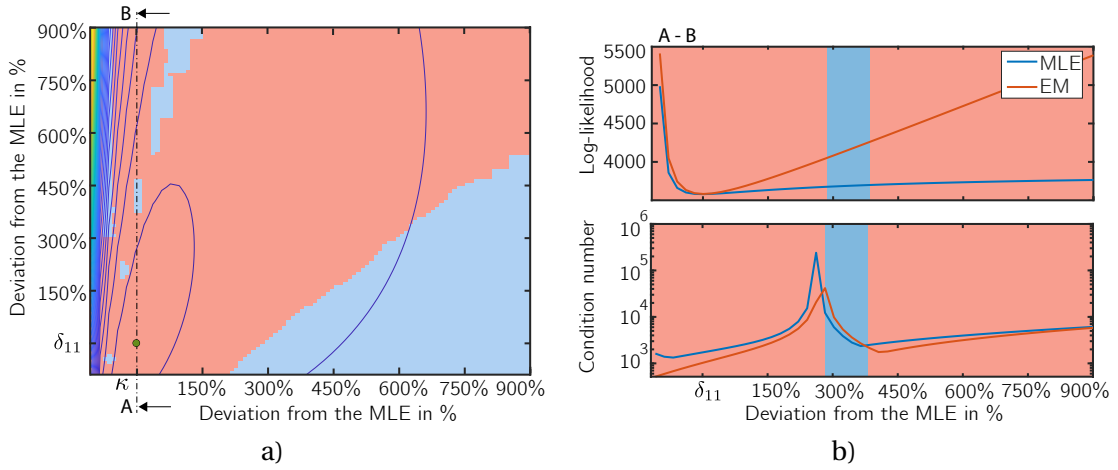


Figure 1.11: Comparison of condition numbers for Hessian matrices. a) Red tiles represent a better-conditioned Hessian for the ECLL, whereas blue tiles represent better-conditioned incomplete log-likelihood. b) Comparison of the condition numbers in κ-direction.

where $\hat{\Theta}_M = \{\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(M)}\}$ denotes the set of estimates coming from M initial values. The best MLE-estimates $\bar{\theta}_{\text{MLE}}$ are then compared to the worst (lowest ECLL) EM-estimates $\underline{\theta}_{\text{EM}}$ for all scenarios. Table 1.4 presents the results for the scenario with the best MLE-performance measured as a relative distance from the reference parameter values. Clearly, even the benefit of $M = 100$ different starting values is not enough to outperform a single run (here, the worst case) of the EM. This puts the runtimes in Table 1.2 into perspective. Despite MLE being significantly faster per single run, a large number of runs is needed in order to ensure convergence to the global maximum.

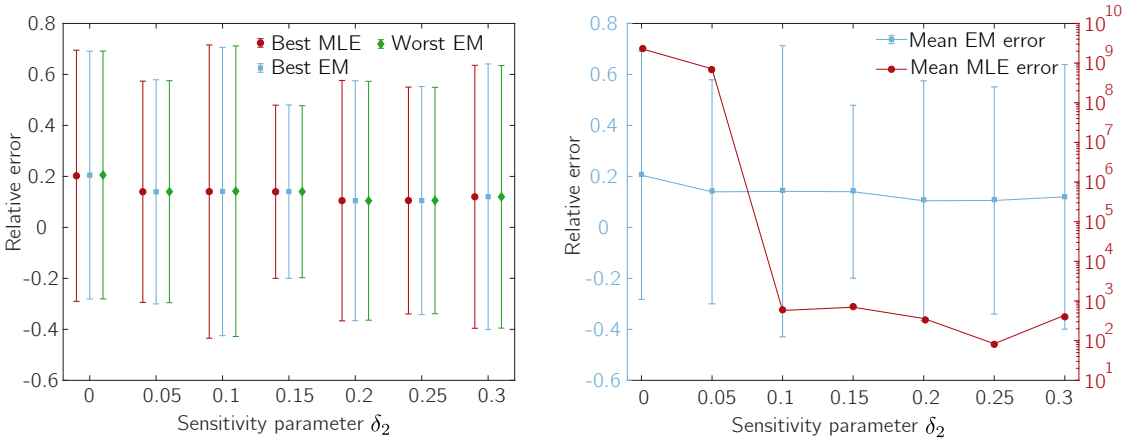
To evaluate the accuracy of the estimation with a single number, we define the (aggregate) relative error for the best (resp., worst) case as:

$$\bar{e} = \frac{\|\bar{\theta} - \theta_r\|}{\|\theta_r\|} \quad \text{and} \quad \underline{e} = \frac{\|\underline{\theta} - \theta_r\|}{\|\theta_r\|}.$$

The evidence from our data indicates that the worst-case and the best-case EM-estimates measured in the complete log-likelihood function value are almost indistinguishable. The difference between highest and lowest value of the complete log-likelihood, over the batch of initial guesses, was on average in the order of 10^{-2}. This means that the sample distribution of the relative error for the worst-case and best-case EM-estimates are almost identical. This dramatically contrasts to the MLE relative-error distribution presented in Fig. 1.9, where the difference between the best and worst estimates can be extremely large. These empirical relationships are captured in Table 1.3. The superior convergence performance of the EM-algorithm and negligible difference in the best to worst comparison advocates for EM as a more robust method of the two. Throughout the numerical experiment *we have not observed a single instance of the direct MLE-procedure outperforming the EM in terms of convergence*. Given the substantial number of scenarios tested, we believe that this is a representative and significant result.

Remark 4 (Action Sensitivity) It is worth pointing out that EM may improve the estimation performance of MLE even in settings with limited significance of the control process $a(t)$.[12] Fig. 1.12 showcases the estimation performance, measured in terms of the relative errors of both estimation methods for various δ_2. We consider a similar setup as in the previous section (i.e., 10 initial guesses for the solver and 10 independently generated portfolios for each value δ_2). In addition, we employ the same filtering technique using the log-likelihood function value to separate the best MLE and the worst EM-estimates. As expected, the relative error of the best MLEs closely corresponds to the EM-estimates; see Fig. 1.12a. However, when considering the average relative error of all solutions (not just the best), we observe the previously recorded behavior of MLE's divergence as a result of disparate local likelihood-minima; see Fig. 1.12b. This suggests that EM can be a preferred estimation method even for

[12] For any given realization of $a(t)$, for $t \geq 0$, the importance of the control process for the evolution of the arrival intensity is fully described by the (nonnegative) sensitivity parameter δ_2. The case of an *autonomous* Hawkes process (*without* control) corresponds to a degenerate situation with $\delta_2 = 0$.



a) b)

Figure 1.12: Impact of sensitivity parameter δ_2 on the relative error. Reported values are averages over 10 independently generated portfolios. a) Relative error of the estimates producing the highest log-likelihood over 10 initial guesses. b) Average relative error of all estimates over 10 initial guesses. Error bands mark the applicable coefficients of variation.

Best of 100		Worst of 100
$\bar{e}_{\text{EM}}$	$\approx$	$\underline{e}_{\text{EM}}$
$\bar{e}_{\text{MLE}}$	$\ll$	$\underline{e}_{\text{MLE}}$
$\bar{e}_{\text{MLE}}$	$\approx$	$\underline{e}_{\text{EM}}$

Table 1.3: Empirical relationship between EM-estimates and MLE-estimates.

uncontrolled (i.e., autonomous) Hawkes processes.

1.5 Conclusion

We have constructed an alternative estimation method for (linear) controlled Hawkes processes based on the EM-algorithm. Compared to conventional maximum-likelihood maximization, the presented method exhibits a substantially more robust behavior in terms of convergence and choice of the initial guesses. The robustness was tested based on extensive

	$\hat{\kappa}$	$\hat{\lambda}_0$	$\hat{\lambda}_\infty$	$\hat{\delta}_{11}$	$\hat{\delta}_{12}$	$\hat{\delta}_{21}$	$\hat{\delta}_{22}$	$\hat{\delta}_{23}$
$\bar{\theta}_{\text{MLE}}$	0.3703	0.0434	0.0312	0.0652	0.0726	0.0238	0.0554	0.0695
Bias	(-7.44%)	(-13.15%)	(+4.12%)	(-18.53%)	(+20.93%)	(-20.74%)	(-7.71%)	(-22.75%)
$\underline{\theta}_{\text{EM}}$	0.3702	0.0434	0.0312	0.0652	0.0725	0.0238	0.0554	0.0695
Bias	(-7.45%)	(-13.14%)	(+4.11%)	(-18.53%)	(+20.91%)	(-20.69%)	(-7.71%)	(-22.76%)

Table 1.4: Comparison of the best MLE-estimate to the worst EM-estimate; results rounded to four significant digits; relative errors (bias) in parentheses.

synthetic credit-collections data mirroring sparse repayment observations as encountered in practice. The EM-algorithm performed reliably well across all scenarios and produced maximum-likelihood estimates with a variance significantly below that produced by the standard MLE-method. The bias of the EM-method was assessed on the best-case MLE to the worst-case EM measured in the value of the log-likelihood function. The difference in the estimates produced was inconsequential (±0.02%) suggesting that the EM-algorithm provides a significant stability gain and would therefore be the advised method for the estimation of linear CHP. Our findings suggest that EM is a viable alternative to the conventional MLE, and in applications where a rich history of observations is unavailable, it is a superior estimation method. In cases where direct maximization is preferred, the EM-algorithm may be used to obtain bootstrapped initial seeds of the model parameters in question. On the theoretical side, we have shown that the (nonnegative) difference between the incomplete log-likelihood and the expected complete log-likelihood (ECLL) is given by the entropy of the branching distribution, thus establishing a lower bound for the incomplete log-likelihood which at the optimal EM-estimator becomes binding.

Acknowledgements

The authors wish to thank Naveed Chehrazi, two anonymous referees, as well as participants of the 2018 INFORMS Annual Meeting in Phoenix, Arizona, for helpful comments and suggestions. Support for this research by the Swiss National Science foundation (under grant no. 105218-179175) is gratefully acknowledged.

Notation

Symbol	Description	Range
$a(t)$	Account treatment schedule	$\mathbb{R}_+$
$g(t,m)$	Triggering kernel	$\mathbb{R}$
$\mathscr{L}(\cdot)$	Incomplete log-likelihood function	$\mathbb{R}$
$\mathscr{L}_C(\cdot)$	Complete log-likelihood function	$\mathbb{R}$
m_i	Relative repayment at time $T=\tau_i$	$\mathbb{R}_+$
$\mathscr{H}_t$	Available information at time t	–
$J(t)=[N(t),R(t)]$	Repayment process	$\mathbb{N}\times\mathbb{R}_+$
$N(t)$	Repayment counting process	$\mathbb{N}$
$Q(\theta,\theta_n)$	Expected complete log-likelihood function	$\mathbb{R}$
$R(t)$	Cumulative relative-repayment process	$\mathbb{R}_+$
r_i	Relative repayment at time $T=\tau_i$	$[0,1]$
t	Current time	$\mathbb{R}_+$
T	Observation period	$\mathbb{R}_{++}$
δ_1	Sensitivity of intensity w.r.t. J	$\mathbb{R}_{++}^{\dim(J)}$
δ_2	Sensitivity of intensity w.r.t. a	$\mathbb{R}_{++}^{\mathbb{N}}$
κ	Mean-reversion parameter	$\mathbb{R}_{++}$
$\lambda(t)$	Intensity process	$\mathbb{R}_+$
λ_0	Initial value of intensity ($\lambda(0)=\lambda_0$)	$\mathbb{R}_{++}$
λ_∞	Long-run stationary value of intensity	$\mathbb{R}_+$
$\mu(t)$	Base-rate intensity	$\mathbb{R}$
θ	Vector of process parameters ($\theta=(\kappa,\lambda_0,\lambda_\infty,\delta_1,\delta_2)$)	Θ
ϑ_j	Time of j-th account-treatment action	$\mathbb{R}_{++}$
τ_i	Arrival time of i-th repayment ($i\geq 1$)	$\mathbb{R}_{++}$

Control Part II

2 Interpretable Reinforcement Learning in Credit Collections

The contents of this chapter are inspired by Mark, M., Chehrazi, N., and Weber, T. A. (2020a). Reinforcement-Learning Approach to Credit Collections. Working Paper, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, presented at INFORMS 2020 Annual Meeting for the Best Student Paper Competition in Finance where it secured first place.

2.1 Introduction

This paper introduces a reinforcement-learning agent that optimizes the process of collecting outstanding (unsecured) consumer credit balances. A debtor's repayment behavior, characterized by a stochastic self-exciting point process developed by Chehrazi and Weber (2015), specifies the timing and the magnitude of random repayments. In this setting, a collector can optionally administer a costly account treatment, which temporarily increases the intensity (the probability of repayment) of the repayment process. The collections problem is then formulated in terms of the stochastic optimal control of the conditional repayment intensity, by means of a dynamic account-treatment policy (characterized by the optimal times and types of particular account-treatment interventions). We provide a Markov decision process (MDP) formulation of the problem and, using a combination of several state-of-the art reinforcement-learning results, construct a highly performing deep-learning agent. From this perspective, our learning agent can be viewed as a purely data-driven controller (i.e., without specifying any analytical details of the process dynamics) of a univariate self-exciting point process. Furthermore, the general framing of the problem as an optimization of agent-environment interactions allows for an out-of-the-box application to other, more complex models of repayment dynamics. There are three main contributions. Firstly, in contrast to other contemporary reinforcement-learning applications, our problem features an asynchronous feedback-reward relationship, which is rarely studied—despite being present in numerous applications (e.g., human learning). This dramatically complicates learning, as the learner is effectively unable to match an action to its probabilistic (and delayed) reward (so called credit assignment problem). To this end, we formulate a reward shaping theorem that distributes otherwise discretely perceived reward signals continuously per every step in an episode, hence circumventing the

credit attribution problem and enabling the learning process. We demonstrate the validity of this result on the collections application in question; however, it is directly applicable to other problems with dynamics driven by self-exciting point processes. In contrast to the current state of the art for dealing with asynchronous MDPs which is based on a complex LSTM net architecture (Upadhyay et al., 2018), our proposed solution is significantly simpler and works well with plain feed forward neural networks typically used in deep q-learning. Secondly, with respect to the contemporary discussions around interpretable and ethical machine learning models, we formulate a domain knowledge regularizer which penalizes the learner so as to adhere to a priori specified structural constraints, and thus delivers understandable and interpretable decision rules. The principal objective is to contribute to the reduction of the "lawlessness of machine learning algorithms', hence guarantee basic audibility of learned decision rules. Finally, we provide a linear approximator based on B-Splines state-space features, which is capable of delivering a comparable performance while keeping the number of learnable parameters low. Additionally, thanks to being linear in trainable weights, a number of convergence results are directly applicable.

2.1.1 Credit Collections

The problem of credit collections as an operations research problem was broached by Mitchner and Peterson (1957). In their seminal contribution, the authors formulate an optimal stopping problem for determining the best pursuit times for various delinquent debt accounts at Bank of America. Despite their method leading to tangible improvements in the profits, the collection problem remained dormant for the next decade. Revisiting the problem, Liebman (1972) frame the problem as an MDP with transitional probabilities based on various account characteristics. Their solution then relies on the value-iteration method, which quickly reaches its computational limits as more granular account information is added. No further progress on this topic was made until recently, when Abe et al. (2010) modeled the collections process as a constrained MDP which explicitly takes business, legal, and resource constraints into account. Most importantly, the authors provide a dataset of accounts subjected to this new approach while reporting its competitive performance. The same methodology is then discussed by Miller et al. (2012) who focus on the actual implementation of the system for collecting delinquent taxes in the state of New York. Finally, Chehrazi and Weber (2015) describe a dynamic repayment model for the accounts placed in collections. The model is based on a self-exciting point process which incorporates account-specific information such as the holder's FICO score or his annual income, as well as time-varying macroeconomic covariates. In a follow-on paper, Chehrazi et al. (2019) leverage the same model in order to construct an account-treatment schedule which maximizes an account's present value net of collection costs. The account-treatments are proxied with collection effort (measured in hours spent on an account). This contrasts with our approach of directly mapping the collector's available actions onto intensity impulses, which if paired with a fixed cost per action would render the original problem analytically intractable. Additionally, despite using the same model for the repayment behavior, our data-driven policy-construction method is readily extensible to more

complex repayment dynamics, including entirely data-driven environments.

2.1.2 Outline

The paper proceeds as follows. In Section 2.2, we introduce a model for the repayment behavior, cast it into a Markov decision process and lay out the credit-collection problem. In Section 2.3, we construct two benchmark collection policies and a deep reinforcement-learning q-agent. Section 2.4 compares the agents in terms of various performance metrics and examines the learning agent's properties relevant for collection practice. Section 2.5 concludes.

2.2 Model

For the remainder of this section we consider a single delinquent account with outstanding balance $w_0 > 0$, placed in collections at time $t = 0$.[1] The account is credited with random repayments of relative size z_i at random times τ_i until the balance is recovered in full.

2.2.1 Repayment Process

As in Chehrazi et al. (2019), we assume that the repayment behavior $(\tau_i, z_i)_{i\geq 0}$ is described by a controlled Hawkes process with an intensity given by a mean-reverting stochastic differential equation (SDE),

$$d\lambda(t) = \underbrace{\kappa(\lambda_\infty - \lambda(t))dt}_{\text{mean-reversion}} + \underbrace{\delta_1^\top dJ(t)}_{\text{self-excitation}} + \underbrace{dA(t)}_{\text{collection strategy}}, \quad t \geq 0. \tag{2.1}$$

The dynamics in SDE (2.1) are derived from a continuous-time hidden Markov model where an account holder can be in one of two distinct states, "H" or "L." An account holder in state "H" would make random partial repayments at higher frequency than if he was in state "L." The state of the account holder evolves according to a generic Markov dynamics model and can also be influenced by the credit-issuer through costly collection actions. The account holder's state, however, is not observable by the collector, but he can estimate the likelihood that an account holder is in state "H" or "L" using observed past repayments. The Bayesian dynamics of these estimates translate to the SDE specification in Eq. (2.1). In particular, the self-excitation term captures a discrete upward adjustment in the collector's beliefs upon observing a repayment. The jump is positive, since a repayment is more likely in state "H" than in state "L." In Eq. (2.1), the vector $J(t) = [N(t), Z(t)]^\top$ consists of an unmarked counting process $N(t) = \sum_i \mathbb{1}_{\{\tau_i \leq t\}}$ and its marked counterpart $Z(t) = \sum_i z_i \mathbb{1}_{\{\tau_i \leq t\}}$ with marks drawn from an empirically identifiable distribution F_z whose support is included in $[z_{min}, 1]$ with minimum relative repayment $z_{min} > 0$. Conceptually, the former represents the holder's

[1] A credit account is considered *delinquent* if it misses a repayment deadline on its outstanding balance by a prespecified time period (e.g., 30 days).

ability-to-repay while the latter captures his willingness-to-repay. The vector $\delta_1^\top = [\delta_{10}, \delta_{11}]$ describes the sensitivity of the process to repayment events.

In the absence of a repayment, the effective rate of repayment $\lambda(t)$ declines, since a period of inactivity is more likely in state "L" than state "H." This is captured in Eq. (2.1) by the first term where the parameter λ_∞ determines the steady-state of the effective repayment intensity and κ determines the rate of convergence. The latter parameter also determines the covariance properties of the process and can be interpreted in terms of how much "memory" the system retains.

Unlike $\lambda(t)$, the dynamics of the outstanding balance $w(t)$ are relatively simple. At any repayment time τ_i, the account's outstanding balance $w(\tau_i)$ diminishes by the amount repayed, i.e., $w(\tau_i) = (1 - z_i)\, w(\tau_i^-)$. Hence, we have

$$w(t) = w(\tau_i), \qquad \tau_i \le t < \tau_{i+1}. \tag{2.2}$$

Lastly, in the absence of a collection strategy $A(t)$, the Markovian nature of the process allows for a compact representation of the intensity flow,

$$\lambda(s) = \varphi(s, \lambda(t)) = \lambda_\infty + (\lambda(t) - \lambda_\infty) e^{-\kappa s}, \quad s \ge t, \tag{2.3}$$

which describes the law of motion for the intensity starting at $\lambda(t)$, provided no repayments are received on the interval $[t, s]$.

2.2.2 The Collection Problem

As indicated earlier, to increase the probability of a repayment, the credit-issuer (or a collector) can take costly collection actions. We assume that the collector is endowed with $M \ge 2$ account-treatment actions (intervention types), each carrying a different temporary impact on the account's repayment-arrival intensity. Indeed, the M nontrivial account treatments can range from mild actions, such as sending a letter of notice (low impact / low cost) to more extreme measures such as filing a lawsuit (high impact / high cost). An action taken does not guarantee a repayment, but rather increases the probability of receiving one in the near future (since it can change the account state from "L" to "H"). Practically, all available collector actions are mapped onto different intensity impulses in the set $\delta_2 = \{\delta_{2,0}, \delta_{2,1}, \delta_{2,2}, \ldots, \delta_{2,M}\} \subset \mathbb{R}_+$, where $\delta_{2,k-1} < \delta_{2,k}$ for $k \in \{1, \ldots, M\}$ with $\delta_{2,0} = 0$ (the magnitude $\delta_{2,k}$ of intervention type k representing its effectiveness). Therefore, any collection strategy can be encapsulated by a non-decreasing, left-continuous process

$$A(t) = \sum_{j=1}^{\infty} \delta_{2,l(\vartheta_j)} \mathbb{1}_{\{\vartheta_j < t\}}, \quad t \ge 0, \tag{2.4}$$

where the mapping $l : \mathbb{R}_+ \to \{1, \ldots, M\}$ with $\vartheta_j \mapsto l(\vartheta_j)$ describes the type of the action taken at time ϑ_j. Given a collection strategy A and an initial intensity $\lambda(0) = \lambda_0$, the solution to Eq. (2.1)

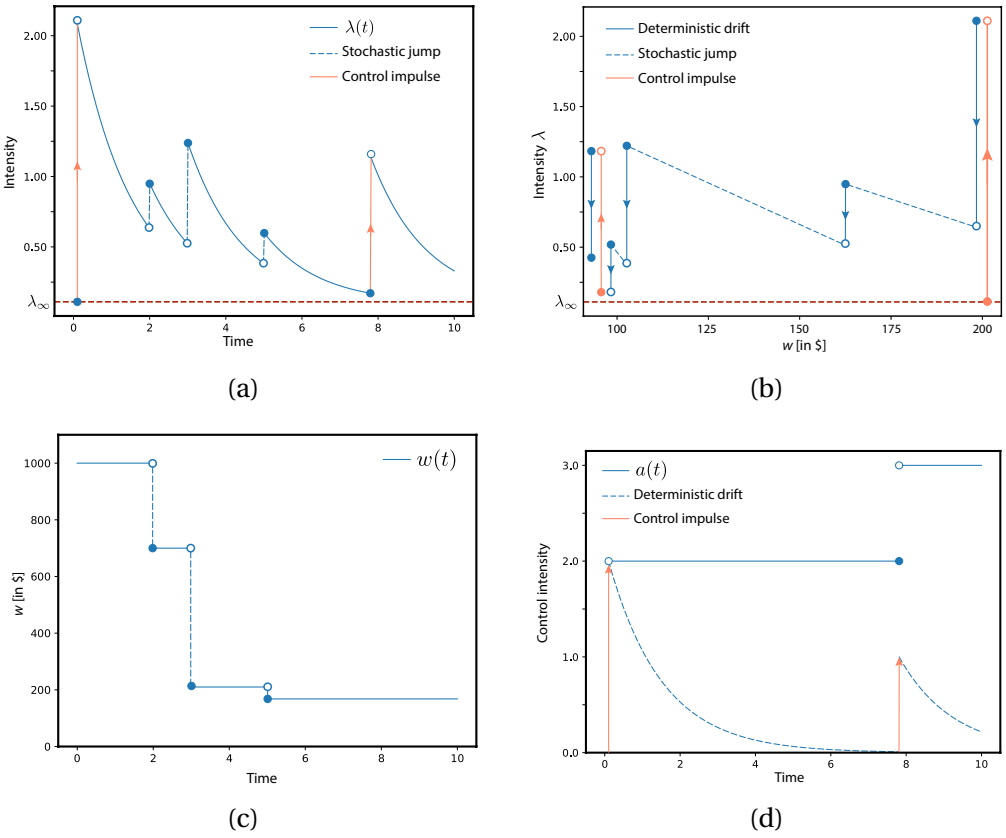


Figure 2.1: Evolution of intensity (a), balance (c) and collection policy (d) in time. The account path through the state space is in subfigure (b). The collection policy is characterized by two actions at time ϑ_1=0.1 and $\vartheta_2 = 7.9$ of respective magnitudes $\delta_{2,1} = 2$ and $\delta_{2,2} = 1$.

can be written as

$$\lambda(t) = \mu(t) + \sum_{i:\tau_i \le t} (\delta_{10} + \delta_{11} z_i) e^{-\kappa(t-\tau_i)} + \sum_{\vartheta_j:\vartheta_j < t} \delta_{2,l(\vartheta_j)} e^{-\kappa(t-\vartheta_j)}, \tag{2.5}$$

where the first term represents a deterministic drift, $\mu(t) = \lambda_\infty + (\lambda_0 - \lambda_\infty)e^{-\kappa t}$; the second term (termed the "endogenous kernel") encapsulates the self-excitation property of the process; and, the last term (termed the "exogenous kernel") features the collection strategy. Additionally, the process history $\mathscr{H}_t$ is captured by the σ-algebra generated by repayment times and amounts up to and including time t as well as account-treatment times and their respective sizes up to but excluding time t. Figs. 2.1a, 2.1c, and 2.1d depict typical paths of intensity, balance, and collector interventions (by means of control impulses) for an account in collections (here, with $(\lambda_0, w_0) = (0.1, \$1000)$). Additionally, Fig. 2.1b showcases the account evolution as a path in the (λ, w) state space. A state-space representation of account evolution and collector intervention is critical for the effective construction of optimal collection policies.

Assuming a linear cost of collection with marginal cost $c > 0$, the *collection problem* boils down

to finding an optimal collection strategy $A^\star$, such that

$$A^\star \in \arg\max_{A\in\mathscr{A}} \mathbb{E}\left[\int_0^\infty e^{-\rho s} w(s^-)\, dZ(s) - c\int_0^\infty e^{-\rho s} dA(s)\,\middle|\, (\lambda(0), w(0))\right], \tag{2.6}$$

where the parameter $\rho \in \mathbb{R}_{++}$ denotes a discount rate, and $\mathscr{A}$ is the space of $\mathscr{H}_t$-predictable collection strategies.

2.2.3 Agent-Environment Interface

To cast the collections problem into a reinforcement-learning framework, the continuous-time Markovian dynamics in Eqs. (2.2), (2.4), and (2.5) must be expressed as a discrete-time Markov chain. In particular, measuring time in small discrete steps of Δt, we assume—without loss of generality—that actions are taken at the beginning of an interval $[k\Delta t, (k+1)\Delta t]$ while repayments, if they occur, are received at the end of such an interval. In fact, this assumption is required to make the discrete-time repayment process non-predictable. From the Poisson dynamics of the repayment process, the likelihood of receiving a repayment at the end of the interval $[k\Delta t, (k+1)\Delta t]$, given initial intensity $\lambda(k\Delta t)$ and action $\delta_{2,l(k\Delta t)}$, is

$$\mathbb{P}[N\big((k+1)\Delta t\big) - N(k\Delta t) = n|\mathscr{H}_{k\Delta t}] = \begin{cases} 1 - \big(\lambda(k\Delta t) + \delta_{2,l(k\Delta t)}\big)\Delta t + o((\Delta t)^2), & n = 0, \\ \big(\lambda(k\Delta t) + \delta_{2,l(k\Delta t)}\big)\Delta t + o((\Delta t)^2), & n = 1, \\ o((\Delta t)^n), & n \geq 2. \end{cases} \tag{2.7}$$

In the previous equation, the discrete-time dynamics of $\lambda(k\Delta t)$ for $k \in \mathbb{Z}_+$ are as follows:

$$\lambda(k\Delta t) = \varphi(\Delta t, \lambda\big((k-1)\Delta t\big) + \delta_{2,l((k-1)\Delta t)}) + (\delta_{10} + \delta_{11} z_{k-1})\mathbb{1}_{\{N(k\Delta t) - N((k-1)\Delta t)\neq 0\}}, \tag{2.8}$$

with $\lambda(0) = \lambda_0$, where we are allowed to use Eq. (2.3), since no discrete event will take place during $(k\Delta t, (k+1)\Delta t)$. Finally, z_k, for $k \in \mathbb{Z}_+$, are independent and identically distributed (i.i.d.) draws from the relative-repayment distribution F_z, so the account balance evolves according to

$$w(k\Delta t) = (1 - z_{k-1})\, w\big((k-1)\Delta t\big)\mathbb{1}_{\{N(k\Delta t) - N((k-1)\Delta t)\neq 0\}}, \quad k \geq 0, \tag{2.9}$$

with $w(0) = w_0$ and $z_{-1} = 0$. Equations (2.7)–(2.9) describe the discrete-time dynamics of the collection process. To simplify the notation, in what follows we denote $\big(\lambda(k\Delta t), w(k\Delta t), \delta_{2,l(k\Delta t)}\big)$ by (λ_k, w_k, a_k). In our numerical implementation, the value of (λ_k, w_k) is quantized a discrete grid on the set of attainable states (λ, w), denoted by $S \subsetneq \mathbb{R}^2_+$. This last step turns the discrete-time, continuous-space Markov dynamics of Eq. (2.7)–(2.9) to a discrete-time finite Markov chain, but otherwise this computational simplification is not critical for our theoretical developments. In particular, it is important to note that in fact we do not restrict attention to the discrete grid of states but rather use it to partition the exploration of the state space to the corresponding subsets (each associated with a corresponding grid point).

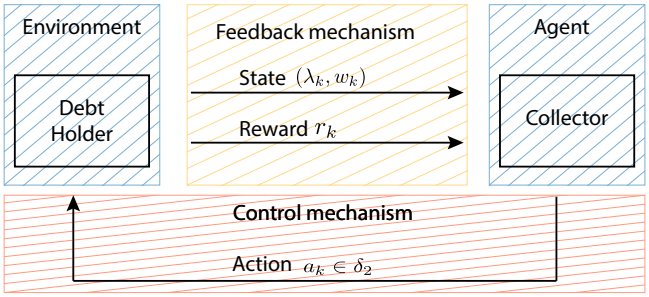


Figure 2.2: Collections process as an MDP.

We can now consider the discrete state-space dynamics, introduced above, as our reinforcement-learning setting. In particular, consider the behavior of the two parties involved: a *decision maker* (also referred to as *agent*) and an *environment* that is responsible for providing feedback on the agent's action in terms of some *reward*.[2] The environment behavior is described by Eqs. (2.7)–(2.9). The agent, following a policy $\pi : \mathscr{S} \to \delta_2$ that prescribes his action for a given state, repeatedly interacts with the environment. At each (discretized) time step $k \geq 0$, the agent observes his state $s_k = (\lambda_k, w_k) \in \mathscr{S}$, selects an action $a_k \in \delta_2$ according to policy $\pi(s_k)$, and the environment responds with the subsequent state $s_{k+1} = (\lambda_{k+1}, w_{k+1})$, together with the reward $r_k \in \mathbb{R}$ observed for going from s_k to s_{k+1}; see Fig. 2.2. The value of the reward r_k is given by

$$r_k = \begin{cases} (z_k w_k - c a_k), & \text{repayment received in } [k\Delta t, (k+1)\Delta t], \\ -c a_k, & \text{no repayment in } [k\Delta t, (k+1)\Delta t]. \end{cases} \tag{2.10}$$

The agent's goal is then to find a solution to the discrete-time collection problem, i.e., find the policy π that maximizes the net collected amount given by

$$v_\pi(\lambda_0, w_0) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k w_k z_k - c \sum_{k=0}^{\infty} \gamma^k a_k \right], \tag{2.11}$$

where $\gamma = \exp(-\rho \Delta t)$.

The preceding specifications are implemented computationally via Alg. 4 (see insert). Equipped with this algorithm, we now have all building blocks for constructing/learning collection policies from a sequence of induced agent-environment interactions.

[2] In engineering applications, the terms *system, controller* and *control signal* are used synonymously for the terms *environment, agent,* and *action* employed here.

Algorithm 1: A discretized simulation algorithm of the repayment process from Eq. (2.1).

Result: Produces a sequence of states s_k for $k \in \{0, 1, \ldots, K\}$, where $w_k \leq \epsilon_w$
Algorithm parameters:
$(\lambda_0, \lambda_\infty, \kappa, \delta_1, \delta_2)$ - process parameters, Δt - discretization step, ϵ_w - balance-error tolerance, π - policy
Initialize the current time $t = 0$, $w_k = w_0, \lambda_k = \lambda_0$
while $w_k > \epsilon_w$ **do**
  Select a according to a policy π, i.e., $a = \pi(s_k)$
  Set $\lambda_k = \lambda_k + a$
  if $\lambda_k \Delta t \geq U[0,1]$ **then**
    Draw a relative repayment z_k according to F_z
    Set $\lambda_k = \varphi(\Delta t, \lambda_k) + \delta_{10} + \delta_{11} z$
    else
      Set $z_k = 0$
      Set $\lambda_k = \varphi(\Delta t, \lambda_k)$
    end
  end
  Set $r_k = (z_k w_k - ac)$
  Set $w_k = (1 - r) w_k$
  Set $k = k + 1$
end

2.3 Collection Policies

Provided the MDP environment description from the previous section, we now design a learning agent capable of finding profitable collection policies in finite time. The performance of a policy is measured, in its simplest form, by the net amount collected, using the value function from Eq. (2.11).

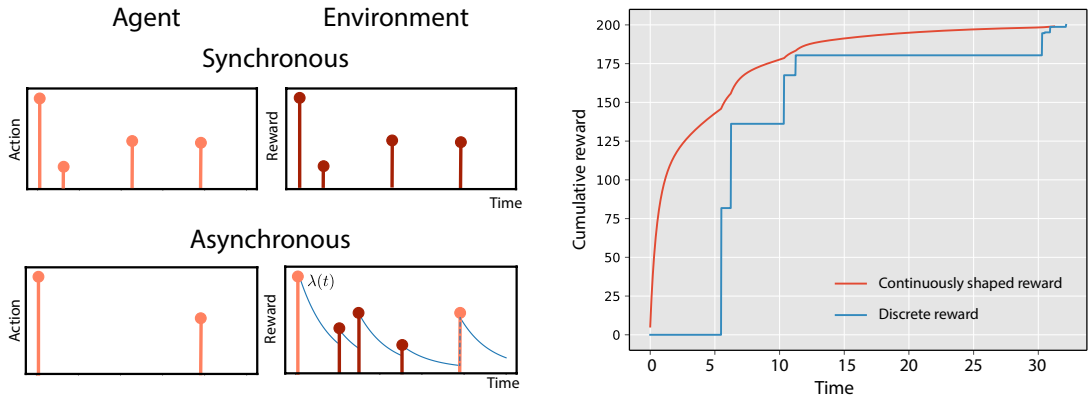
2.3.1 Autonomous Account Value

In the absence of any collection strategy we label the account to be "autonomous," i.e., following the policy $\pi(s) = 0$, for all $s \in S$. The autonomous account value (AAV) is then computed as

$$u(\lambda, w) = \mathbb{E}\left[\sum_{k=0}^{\infty} w_k z_k \gamma^k \middle| w_0 = w, \lambda_0 = \lambda \right], \tag{2.12}$$

which can be easily approximated by Monte Carlo simulation.[3] This quantity serves as a basic benchmark for the agent to beat. Indeed, at the very least, a successful agent should be able to learn to do nothing, that is, to not take collection actions which do not yield tangible improvements over the AAV.

[3] Chehrazi and Weber (2015) provide a quasi-analytical solution to the continuous time analogue of Eq. (2.12).



(a) (b)

Figure 2.3: (a) Difference between synchronous and asynchronous reinforcement-learning setup. (b) Comparison of the continuously reshaped cumulative reward and the original discrete reward formulation.

2.3.2 Optimal Policy under Continuous Collection Effort

Chehrazi et al. (2019) proposed and solved a variation of the collections problem outlined above. Specifically, rather than mapping available collection actions to intensity impulses from δ_2, the authors modeled the collection effort proxied as a number of hours spent on each account. In practice, this means a collector can exert an intensity impulse of size $a \in \mathbb{R}_{++}$, which produces a status change of the account when entering the collection process. Additionally, the collector can maintain the intensity at a specific intensity level $\hat{\lambda}$ via a continuous infinitesimal thrust, which captures the effect of the action while it is active, for instance, until an agreement for a repayment plan is reached. This formulation of the discrete-time collection problem allows for a semi-analytical solution of the value function $v^{\star}_{\mathrm{CE}}(s)$. Given the more restrictive set of controls with only a finite number of discrete actions available and without the possibility to sustain the intensity level, the solution developed by Chehrazi et al. (2019) acts as a natural upper bound for our problem: $v^{\star}_{\mathrm{CE}}(\lambda, w) \geq v^{\star}(\lambda, w)$, for all $(\lambda, w) \in \mathbb{R}^2_+$.

2.3.3 Deep Q-Approach

Q-learning is a model-free learning algorithm developed by Watkins and Dayan (1992), which gained enormous popularity especially in recent years—largely thanks to its successful application in traditionally difficult environments, such as the Atari game suite (Mnih et al., 2013) or the game of Go (Silver et al., 2017). In fact, the trained agents were able to attain superhuman skill levels in both games, something that was previously impossible with the dynamic programming (DP) approach. Contrary to a value-iteration algorithm (Bertsekas, 1987) that relies on learning and subsequent improving of the value function $v_\pi(s_k)$, the q-agent attempts to learn a so-called q-function $q_\pi : \mathcal{S} \times \delta_2 \to \mathbb{R}$ by quantifying the quality of a state-action pair

in terms of its value, i.e., $q_\pi(s_k, a) = \mathbb{E}[r_k + \gamma v_\pi(s_{k+1})|a_k = a]$.[4] In other words a state-action pair represents the maximum expected return achievable by following policy π, after taking an action a at state s_k, so $v_\pi(s_k) = \max_{a\in\delta_2} q_\pi(s_k, a)$.

Recall from DP, the optimal value function $v^\star$, and hence the optimal state-action $q^\star$-function, satisfies the Bellman equation,

$$\begin{aligned} q^\star(s_k, a_k) &= \sum_{s'} P(s'|s_k, a_k)\left[R(s_k, a_k, s') + \gamma \max_{a'} q^\star(s', a')|s_k, a_k\right] \\ &= \mathbb{E}_{s'\sim P(\cdot)}\left[R(s_k, a_k, s') + \gamma \max_{a'} q^\star(s', a')|s_k, a_k\right], \end{aligned} \tag{2.13}$$

where $P(s'|s_k, a_k)$ determines the transition probabilities of the environment and $R(s_k, a_k, s')$ is a (possibly random) reward encountered when going from s_k to s' (denoted by $s_k \to s'$) after selecting action a_k at time step k. For discrete state-space problems, with completely known system dynamics, finding the $q^\star$-function leads to solving a linear system of $|\mathscr{S}| \times |\delta_2|$ equations. Therefore, the $q^\star$-function would reduce to a simple look-up table with rows representing the attainable states and columns corresponding to all available actions. Practically, it is often preferred to solve Eq. (2.13) iteratively, as a fixed point problem resulting in the value iteration algorithm. That is, using the Bellman backup operator c it is

$$\mathscr{T} q(s_k, a_k) \triangleq \mathbb{E}_{s'\sim P(\cdot)}\left[R(s_k, a_k, s') + \gamma \max_{a'\in\delta_2} q(s', a')\right], \tag{2.14}$$

and starting from some seed mapping $q_0 : \mathscr{S} \times \delta_2 \to \mathbb{R}$ we define a sequence of approximate q-functions $\{q_i\}_{i\in\mathbb{N}}$ obtained as $q_{i+1} = \mathscr{T} q_i$ which is proved to converge to the optimal $q^\star$ as $i \to \infty$ (Bertsekas, 2011).

Finally, for environments with unknown dynamics, we often approximate the expectation with temporal-difference (TD) updates which form the basis for the q-learning algorithm,

$$q_{i+1}(s_k, a_k) \triangleq q_i(s_k, a_k) + \alpha_k\left(r_k + \gamma \max_{a'} q_i(s_{k+1}, a') - q_i(s_k, a_k)\right), \tag{2.15}$$

where $\alpha_k > 0$ represents the (time-dependent) learning rate and $\gamma \in (0, 1)$ the discount factor. Similar to DP, the estimates are updated with previously learned estimates, i.e., q-learning relies on bootstrapping. The agent then repeatedly interacts with the environment forming complete trajectories (episodes) $(s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{\text{terminal}})$, and after each transition updates his q-value estimate of the visited state-action pair.[5] Furthermore, to prevent an agent from learning sub-optimal policies and repeatedly *exploiting* the same action (typically at any time step at least one action yields the highest value), a technique forcing random *exploration*, called ϵ-greedy policy, is applied that guarantees $\mathbb{P}[\pi(s) = a] > 0$ for all $(s, a) \in \mathscr{S} \times \mathscr{A}$ (Sutton

[4] The subscript π designates the policy that is being followed by the agent.

[5] A state s_k is considered "terminal" (denoted by s_{terminal}) if a balance w_k falls below some tolerance ϵ_w.

and Barto, 1998). Lastly, under some mild conditions on the learning rate and provided that the state and action spaces are finite, the q-learning algorithm from Eq. (2.15) is guaranteed to converge to an optimal solution (Sutton, 1988; Melo, 2001).

In reality, this look-up table approach is often highly impractical, as the state-action pair is estimated separately for each sequence, without any generalization. Indeed, when taking an action from a given state s, the information about the obtained reward is also relevant for all states in the neighborhood of s. Accordingly, to render the q-learning compliant with the collection problem, which *per se* features an infinite state space with $(\lambda, w) \in \mathbb{R}^2$, we define an approximate q-function $\hat{q}_\pi(s, a; \mathbf{w})$, parametrized by a weight vector $\mathbf{w} \in \mathbb{R}^d$. Instead of a look-up table $q_\pi(s_k, a)$, the q-function is now approximated using a functional form, so $\hat{q}_\pi(s, a; \mathbf{w}) \approx q_\pi(s, a)$. The idea behind using such a parametrization is that the number of weights (the dimensionality of $\mathbf{w}$) is much smaller than the number of states, i.e., $d \ll |\mathscr{S}|$. Typically, the approximators were defined as a linear function of states (linear approximators), as most of the convergence guarantees were directly applicable (Tsitsiklis and Van Roy, 1997; Melo and Ribeiro, 2007; Carvalho et al., 2020). Note that the lookup table is a special case of a linear approximator with a separate weight for each possible state. However, with the advances in the machine-learning field non-linear approximators, such as neural networks, became the favored choice.[6]

Instead of updating the q-values directly at each learning step, we aim to update the weight vector $\mathbf{w}$, to optimize an accuracy metric like Mean Square Error (MSE) or Huber loss (Huber, 1992). Consequently, finding the optimal approximator becomes a regression problem in the form of an iterative minimization of loss functions $\mathscr{L}(\mathbf{w}_k)$,

$$\mathscr{L}(\mathbf{w}_k) = \frac{1}{2} \sum_{s \in \mathscr{S}} \sigma_\pi(s) \sum_{a \in \delta_2} \mathbb{P}[\pi(s) = a] \left[(y_k(s, a) - \hat{q}(s, a; \mathbf{w}_k))^2 \right]. \tag{2.16}$$

In this, $y_k(s, a) = \mathscr{T} \hat{q}(s, a, \mathbf{w}_{k-1})$ is the target for a state-action pair (s, a) at iteration k, and $\sigma_\pi(s)$ denotes the distribution of states under the learned policy π termed *on-policy distribution*. Conceptually, $\sigma_\pi(s)$ is a state distribution (i.e., $\sigma_\pi(s) \geq 0$, $\sum_s \sigma_\pi(s) = 1$) that represents how much we care about the error in each state s. Often, $\sigma_\pi(s)$ is chosen to be the fraction of time spent in s (Sutton and Barto, 1998). In continuing tasks (tasks without an end state), the on-policy distribution is the stationary distribution under π.

We start with a possibly randomly initialized set of weights $\mathbf{w}$ and at each iteration we apply a gradient update,

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \nabla_{w_k} \mathscr{L}(\mathbf{w}_k). \tag{2.17}$$

Rather than computing the gradient over the full expectation of Eq. (2.16), it is often computationally more efficient to make an approximation using real samples (experience) from the on-policy distribution. In practice, the training of the q-function relies on the experience

[6] For a comprehensive review of other commonly used approximators we refer readers to (Ch. 9, Sutton and Barto, 1998).

Algorithm 2: Deep Q-agent with experience replay and guided exploration.

Algorithm parameters:
$(\lambda_0, \lambda_\infty, \kappa, \delta_1, \delta_2)$ - process parameters, Δt - discretization step, π - policy,
N_{episodes} - number of episodes
Initialize the replay buffer D to some fixed size N
Initialize a neural network with a starting set of weights $\mathbf{w}_0$
for *episode=1:*$N_{episodes}$ **do**
  Select a starting state $s_0 = (\lambda_0, w_0)$ according to the guided-exploration rule
  Set $k = 0$ **while** s_k *is not terminal* **do**
    Select a random action a_k with probability ϵ, otherwise
     $a_k \in \arg\max_a q(s_k, a; \mathbf{w}_k)$
    Take an action a_k, observe reward r_k, next state s_{k+1} and a Boolean flag
     indicating whether s_{k+1} is terminal state or not
    Store the transition $\tau_k = (s_k, a_k, r_k, s_{k+1})$ in the experience replay D
    Sample a random minibatch of transitions τ_l from the buffer D
    Set $y_l = \begin{cases} r_l & \text{for terminal } s_{l+1} \\ r_l + \gamma \max_a q(s_l, a; \mathbf{w}) & \text{for non-terminal } s_{l+1} \end{cases}$
    Perform a gradient-descent step
     $\mathbf{w}_{k+1} \in \arg\min_{\mathbf{w}_k} \mathbb{E}_{\tau_l \sim D} \left[\left(r_l + \gamma \max_{a \in \delta_2} q(s_{l+1}, a, \mathbf{w}_{k-1}) - q(s_l, a, \mathbf{w}_k) \right)^2 \right]$
  end
end

replay buffer, which allows a reinforcement-learning agent to store experience (trajectories of the MDP) in the form of transition tuples denoted $\tau_k = (s_k, a_k, r_k, s_{k+1})$. At every training step, these are then randomly sampled, forming a training mini batch for the stochastic gradient-descent optimization. The intuition behind memory replay is to break extant temporal correlations among observations, and thus stabilize the learning (Liu and Zou, 2018). The gradient based of the loss functions is then given by

$$\nabla_{\mathbf{w}_k} \mathscr{L}(\mathbf{w}_k) = \mathbb{E}_{\tau_k \sim D} \left[\left(r_k + \gamma \max_{a'} \hat{q}(s_{k+1}, a'; \mathbf{w}_{k-1}) - \hat{q}(s_k, a_k; \mathbf{w}_k) \right) \nabla_{\mathbf{w}_k} \hat{q}(s_k, a_k; \mathbf{w}_k) \right], \quad (2.18)$$

where τ_k are transition tuples from the replay buffer D, which is designed to store samples for the past N transitions. Our implementation employs an adaptation of the uniform memory buffer coined Prioritized Experience Replay (PER); for details, see Appendix A.1.

Q-Function Approximation

Modern reinforcement learning applications often employ deep neural networks as q-function approximators due to their ability to capture "arbitrarily" complex patterns via universal approximation theorem (Cybenko, 1989). In this text we consider two types of approximators,

deep ReLU neural network as a nonlinear approximator and a shallow B-Spline network as a linear approximator.

Generally, a feedforward ReLU network $f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_{L+1}}$ is a superposition of L hidden layers $h_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$, each of width $\{d_i\}_1^L \subseteq \mathbb{N}$, i.e.,

$$f(x) = \mathbf{W}_{L+1}\phi(\mathbf{W}_L\phi(\mathbf{W}_{L-1}\dots\phi(\mathbf{W}_2\phi(\mathbf{W}_1 x + b_1) + b_2)\cdots + b_{L-1}) + b_L) + b_{L+1}, \tag{2.19}$$

for all $x \in \mathbb{R}^{d_0}$ where $\mathbf{W}_l \in \mathbb{R}^{d_i \times d_{i-1}}$ and b_i is the matrix of trainable weights and the vector of biases for the i-th layer respectively, and $\phi(x) = \max\{0, x\}$ is the rectified linear unit activation. In our application, the input dimension d_0 corresponds to the state-space dimensions, whereas the output dimension d_{L+1} is equal to the number of actions available. Fig. 2.4a depicts a sample architecture of a q-approximator with three available actions. Computing the gradient of Eq. (2.19) is straightforward with the aid of the backpropagation algorithm.

Additionally, we consider a linear q-function approximator based on Basis spline (B-spline) features. Technically, we reformulate the minimization problem from Eq. (2.16) as a surface spline regression, that is, a linear regression with features defined by a tensor product of B-splines of order three. A set of n B-spline basis functions $\{B_{i,p;\mathbf{t}}\}_{i=1}^n$ of degree p is defined using the de Boor's recursion formula as

$$B_{i,p;\mathbf{t}}(x) = \frac{x - t_i}{t_{i+p} - t_i} B_{i,p-1;\mathbf{t}}(x) + \frac{t_{i+p+1} - x}{t_{i+p+1} - t_{i+1}} B_{i+1,p-1;\mathbf{t}}(x),$$
$$B_{i,0;\mathbf{t}} = \begin{cases} 1 & \text{if } t_i \le x \le t_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

for all $x \in \mathbb{R}$ where a knot vector $\mathbf{t} = \{t_i\}_{i=1}^{n+p+1}$ is a monotonically increasing sequence of points. In our application, we define a q-approximator as a tensor product surface defined by two uniformly spaced knot vectors $\mathbf{t}_\omega = (\omega^{(0)}, \dots, \omega^{(r)})$ and $\mathbf{t}_\lambda = (\lambda^{(0)}, \dots, \lambda^{(q)})$ of respective lengths $r+1$ and $q+1$, with $\omega^{(0)} = 0$, $\omega^{(r)} = w_{\max}$, $\lambda^{(0)} = \lambda_\infty$ and $\lambda^{(q)} = \lambda_{\max}$. Specifically, for each fixed action $a \in \delta_2$ the approximate action-value function is defined as

$$\hat{q}(s, a; \mathbf{w}) = \sum_{i=0}^{m} \sum_{j=0}^{n} \omega_{i,j} B_{i,3;\mathbf{t}_\lambda}(\lambda) B_{j,3;\mathbf{t}_\omega}(w), \tag{2.20}$$

where $s = (\lambda, w)$, $\mathbf{w} = \{\omega_{i,j}\}_{(0,0)}^{(m,n)}$, $m = q - 4$, and $n = r - 4$. For a thorough analysis of B-spline properties the reader is referred to Dierckx (1995). Finally, thanks to being linear in weights, we can directly apply Carvalho et al. (2020) convergence result for off-policy trained q-learning with linear function approximators. A sample architecture of the B-spline approximator is demonstrated in Fig. 2.4.

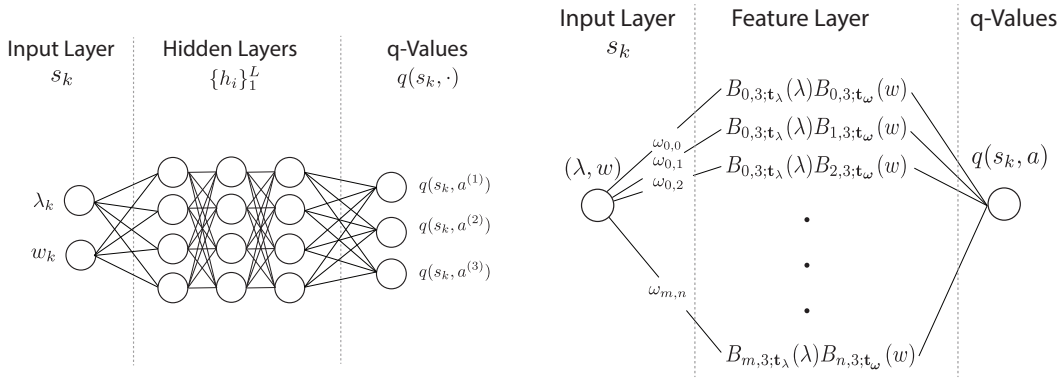


Figure 2.4: a) Neural network architecture of the q-function. b) Architecture of the B-spline surface approximator.

Continuous Reward Shaping[7]

The complexity of the problem of maximizing the objective in Eq. (2.11) arises mainly from an asynchronous relationship between an action and its resulting reward. In contrast to a delayed reward environment (Campbell et al., 2014) where each reward is stochastically delayed by times drawn from a known family of stationary distributions, the collections problem is significantly more intricate. The agent's actions are asynchronous to the environment's discrete feedback; see Fig. 2.3. Despite this formulation being natural to many practical applications, the vast majority of applications considers synchronous actions and feedback with the notable exception of Upadhyay et al. (2018).

In reinforcement-learning problems with sparse rewards similar to the collection case, the inherent difficulty is that the agent does not receive the learning signal sufficiently often to learn from, and even if he does, it is usually very far (in time) from the actions that led to this reward. Learning in such environments often takes much longer time or is outright impossible. A possible remedy is to *shape* the rewards by providing some auxiliary small reward that nudges the agent's learning into the "right" direction. In practice, the auxiliary reward is designed by a human expert who tries to encourage some positively perceived behavior (e.g., in the game of Pong, it is common to provide an agent with a small positive reward for every step the ball is traveling towards the opponent's paddle). In our case, we leverage the martingale property of compensated point processes to reshape discrete rewards into a continuous reward stream.

[7] The idea presented in this subsection is due to Naveed Chehrazi and is further discussed in Mark et al. (2020a).

Proposition 1 The sparse reward r_k obtained at step k

$$r_k = -ca_k + \begin{cases} z_k w_k, & \text{repayment received in } [k\Delta t, (k+1)\Delta t], \\ 0, & \text{otherwise.} \end{cases} \tag{2.21}$$

is in expectation equal to $r_k = w((k-1)\Delta t)\bar{z}\lambda(k\Delta t)\Delta t$, where $\bar{z}$ is the mean relative repayment size, Δt is the environment's discretization step, and $\big(\lambda(k\Delta t), w(k\Delta t)\big)$ represents the account state at step k.

Proof. Proof of Proposition 1. As a consequence of Eq. (2.6), the value of an account following a policy π is given by

$$v_\pi(\lambda, w) = \mathbb{E}_\pi\left[\int_0^\infty e^{-\rho s} w(s^-)\, dZ(s) - c\int_0^\infty e^{-\rho s}\, dA(s)\,\Big|\,\mathscr{H}_0\right].$$

Taking advantage of the martingale identity,

$$\mathbb{E}\left[\int_0^t e^{-\rho s} w(s^-)\, dZ(s)\,\Big|\,\mathscr{H}_0\right] = \mathbb{E}\left[\int_0^t e^{-\rho s} w(s^-)\bar{z}\lambda(s)\, ds\,\Big|\,\mathscr{H}_0\right], \qquad t \in \mathbb{R}_+,$$

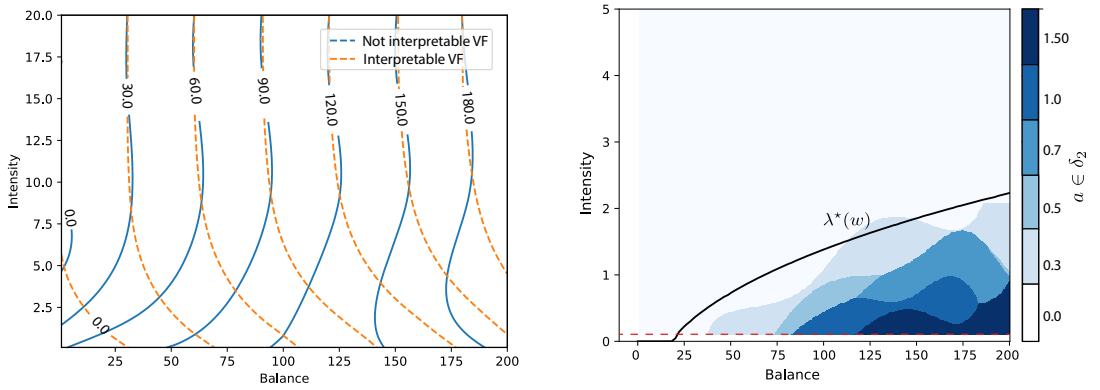
where $\bar{z} = \mathbb{E}[z_k]$, the account value can be rewritten as

$$v_\pi(\lambda, w) = \mathbb{E}_\pi\left[\int_0^\infty e^{-\rho s} w(s^-)\bar{z}\lambda(s)\, ds - c\int_0^\infty e^{-\rho s}\, dA(s)\,\Big|\,\mathscr{H}_0\right],$$

where the terms in expectation represent the discounted accumulated reward and cost of collections, respectively. Discretizing the first integral and omitting the discounting term yields the continuous one-step reward $r_k = w((k-1)\Delta t)\bar{z}\lambda(k\Delta t)\Delta t$, analogous to the sparse one-step reward in Eq. (2.21), which concludes the proof. For a comparison of the sparse and continuous reward formulations, see Fig. 2.3.

Regularizer-Fitted Q-Learning

Despite major breakthroughs, reinforcement learning has not yet been widely adopted for business decision making problems (Dulac-Arnold et al., 2019; Ghassemi et al., 2020). The principal cause is twofold. Firstly, RL algorithms are by their very nature extremely data-hungry, and, as such, are applicable only where large amounts of data can be generated on demand (e.g., robotics or model based RL). Secondly, practical applications often impose additional requirements on the policy that go well beyond mere performance metrics, such as *interpretability* of the resulting decision rules and thus their comprehensibility for human decision makers. For instance, when deciding on how much credit to extend to a car-loan applicant, we expect this point estimate to be not only sufficiently accurate, but also monotonically increasing in the applicant's salary and credit rating. However, when training a neural net on real data, despite a favourable loss metric, the sheer amount of learnable weights makes the model susceptible to overfitting, and thus to obscuring such an intuitive and im-



(a) (b)

Figure 2.5: (a) Interpretable vs not Interpretable value function (b) Decision rule violating interpretability in certain regions, $\lambda^\star(w)$ denotes the optimal policy for the case with continuous actions.

portant relationship. The resulting local inconsistencies would tend to undermine decision maker's confidence in the model, which therefore would not stand a good chance of getting implemented. Should the model nevertheless pass the validation phase and be adopted in practice, it is prone to produce locally biased predictions, which would predominantly affect underrepresented subgroups for which the available data are relatively sparse.

For the collections problem in question, the interpretability of a given policy π and its associate value function v_π is closely linked to the structure and the shape of the action set and value function, respectively. Specifically, this structure must follow systemic consistency conditions which can be framed as value function and policy monotonicity constraints. That is, values of accounts for fixed intensity λ need to increase in the outstanding balance w. Similarly, for accounts with fixed balance w, an increase in intensity λ needs to produce larger account values so

$$\big(w' \geq w \;\Rightarrow\; \hat{q}((\lambda, w'), a; \mathbf{w}) \geq \hat{q}((\lambda, w), a; \mathbf{w})\big) \;\wedge\; \big(\lambda \geq \lambda' \;\Rightarrow\; \hat{q}((\lambda', w), a; \mathbf{w}) \geq \hat{q}((\lambda, w), a; \mathbf{w})\big), \tag{2.22}$$

for all $a \in \delta_2$. Additionally, consistent with the economic principle of free disposal of effort it is important that the policy presented to a rational decision maker is that if at any point in the state space (λ, w) a (nonzero) action is prescribed, then actions of at least the same strength need to also be prescribed for intensities lower than λ and outstanding balances higher than w.

These consistency conditions, which imply shape constraints on the value function, capture the economic logic that if it is optimal to act for an account in a lower balance state, then it must also be optimal to act (at least as forcefully) for an account at a higher balance. Similarly, an account in lower intensity state is less likely to repay, so an optimal action has to be at least of the same size. For a detailed analysis of the theoretical properties of policy and value

function, see Chehrazi et al. (2019) who obtain an optimal solution for the collection problem in continuous time. The monotonicity constraints in Eq. (2.22) can be regarded as prior structural knowledge and can be included in the learning by means of a barrier regularization term,

$$H(\hat{q}(s,a;\mathbf{w})) = \eta_1 \max\{0, -\frac{\partial \hat{q}(s,a;\mathbf{w})}{\partial \lambda}\} + \eta_2 \max\{0, -\frac{\partial \hat{q}(s,a;\mathbf{w})}{\partial w}\}. \tag{2.23}$$

Therefore, we define a domain-knowledge regularized (DKR) policy iteration algorithm using a monotonicity-regularized Bellman operator $\mathscr{T}^R$

$$\mathscr{T}^R \hat{q}(s_k,a_k) \triangleq \mathbb{E}_{s_{k+1}\sim P(\cdot|s_k,a_k)}\left[\hat{R}(s_k,a_k,s_{k+1}) + \gamma \hat{v}_\pi(s_{k+1})\right], \tag{2.24}$$

where the penalization for monotonicity is absorbed in an augmented reward term

$$\hat{R}(s_k,a_k,s_{k+1}) = R(s_k,a_k,s_{k+1}) - H(\hat{q}(s_k,a_k;\mathbf{w})). \tag{2.25}$$

The augmented reward notation yields an identical update rule to Eq. (2.14); thus, the sequence of q-functions, defined recursively by $q_{i+1} = \mathscr{T}^\pi q_i$, converges to a regularized q-function as $i \to \infty$. The augmented reward acts as a *soft* penalization of the action-value functions that violate the nonotonicity constraints via the barrier penalization term. A positive penalization of the q-network loss encourages monotonicity while not entirely restricting the weights out of non-monotonic regions. As a result, we expect that such constraint will favor monotonicity without sacrificing the network's flexibility. This idea can be extended to off-policy learning where the DKR loss function from Eq. (2.18) function is modified to

$$\begin{aligned}\nabla_{\mathbf{w}_k}\mathscr{L}(\mathbf{w}_k) =& \mathbb{E}_{\tau_k)\sim D}\Bigg[\left(r_k + \gamma \max_{a'} \hat{q}(s_{k+1},a';\mathbf{w}_{k-1}) - \hat{q}(s_k,a_k;\mathbf{w}_k)\right)\times \\ & \left(\nabla_{\mathbf{w}_k}\hat{q}(s_k,a_k;\mathbf{w}_k) - \nabla_{\mathbf{w}_k} H(\hat{q}(s_k,a_k;\mathbf{w}_k))\right)\Bigg],\end{aligned}$$

Note that the monotonicity is imposed only "softly," that is the monotonicity of the q-function is not guaranteed but simply encouraged via the penalization term. The penalization hyperparameters η_1 and η_2 are set to be polynomialy increasing in the iteration number i, hence slowly nudging the parameters towards monotonic q-functions. Lastly, using a linear B-spline the q-approximator has a tangible advantages over the non-linear neural net in a sense that convergence guarantees for the q-learning exist (Bertsekas and Tsitsiklis, 1996; Tsitsiklis and Van Roy, 1997), even for the off-policy case with a uniform replay buffer Carvalho et al. (2020). Additionally, the regularization term featuring the q-function derivatives with respect to state-space variables is readily available in closed-form via a simple relationship between the derivative of a spline function and a B-spline of lower degree.

2.4 Results

The empirical identification of an impulse-controlled Hawkes process is discussed in Chehrazi and Weber (2015) using GMM and Mark and Weber (2020) using an EM-type algorithm. In our analysis, we focus on debt holders with similar characteristics, i.e., with fixed repayment-process parameters from Tab. 2.1. However, we differentiate individual accounts with their starting state $(\lambda_0, w_0) \in \mathbb{R}^2_+$. That is, an account perceived as being of a higher quality will have a higher starting intensity λ_0. To evaluate the performance of the developed learners, we rely on a robust numerical experiment. Consider a collector endowed with six treatment actions, no treatment $\delta_{2,0} = 0$, weak treatments $\delta_{2,1} = 0.3, \delta_{2,2} = 0.5$, moderate treatment $\delta_{2,3} = 0.7$, and severe treatments $\delta_{2,4} = 1.0, \delta_{2,5} = 1.5$, along with a portfolio of 200 accounts, $P_{200} = \{(\lambda_0^{(p)}, w_0^{(p)})\}_{p=1}^{200}$. Applying the techniques described in Section 2.3, the learning is carried out over 20,000 collection episodes, where an episode represents a single fully collected account. To ensure learning of stable policies and sufficient exploration of the state space, it is imperative to ensure slow and steady learning (large number of episodes with small and decreasing learning rate α_k). For evaluation of the learning progress we devise three systematic learning measures linked to our objectives – policy quality, speed of convergence, and value function interpretability.

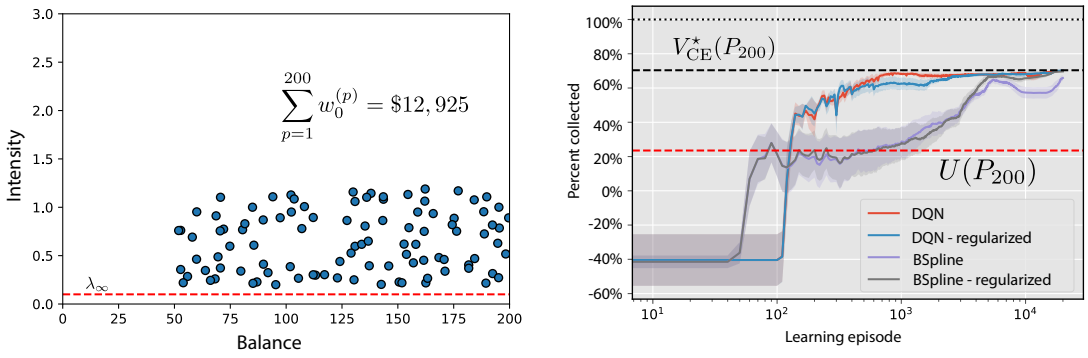
Average reward and regret

Traditionally, the performance of a reinforcement learning algorithm is judged via a visual inspection of its learning curves (i.e., the average reward attained as a function of learning episodes elapsed). For our purposes, we define a variation of this metric given by

$$V(e, P_{200}) = \frac{\sum_{p=1}^{200} \hat{v}(\lambda_0^{(p)}, w_0^{(p)}; \mathbf{w}_e)}{\sum_{p=1}^{200} w_0^{(p)}}. \tag{2.26}$$

That is, a relative amount collected from the entire portfolio at episode e. Additionally, in Sec. 2.3.2 we provided a theoretical optimum (unattainable in our case) of the collection problem with continuous actions that acts as a natural upper bound to our problem, $V(e, P_{200}) \leq V^\star_{\mathrm{CE}}(P_{200} = \frac{\sum_{p=1}^{200} v^\star_{\mathrm{CE}}(\lambda_0^{(p)}, w_0^{(p)})}{\sum_{p=1}^{200} w_0^{(p)}}$. Therefore, we define a natural yet practically unattainable regret of not acting optimally as per the collection policy with continuous and sustain actions as

$$G(e, P_{200}) = \frac{V^\star_{\mathrm{CE}}(P_{200}) - V(e, P_{200})}{V^\star_{\mathrm{CE}}(P_{200})}. \tag{2.27}$$



(a) (b)

Figure 2.6: (a) Portfolio of accounts used for the evaluation of learning metrics. (b) Relative amount collected of the entire portfolio. The red and blacked dashed lines represent the autonomous portfolio value $U(P_{200}) = \frac{\sum_{i=0}^{200} u(\lambda_0^{(p)}, w_0^{(p)})}{\sum_{p=1}^{200} w_0^{(p)}}$ and the theoretical collection bound $V_{\mathrm{CE}}^{\star}(P_{200})$, respectively.

Interpretability index

An interpretable value function $\hat{v}_\pi(s) = \max_{a'} \hat{q}_\pi(s, a')$ needs to satisfy the structural monotonicity constraints with respect to its states as defined in Eq. (2.22). For this sake, we define an interpretability index

$$I = \frac{1}{\lambda_{\max}} \frac{1}{w_{\max}} \int_{\lambda_\infty}^{\lambda_{\max}} \int_{0}^{w_{\max}} \mathbb{1}\left[(\partial_\lambda v(\lambda, w) \geq 0) \wedge (\partial_w v(\lambda, w) \geq 0)\right] d\lambda dw,$$

that computes the number of violations of the monotonicity constraints in percent.

Convergence speed

Finally, the convergence of a RL algorithm can be observed from the plateauing learning curves. We define a learning termination episode E^α for some $\alpha \in \mathbb{R}_+$ such that $\forall e \geq E^\alpha \; G(e, P_{200}) \leq \alpha$.

To demonstrate the stability of the agent-learned policies, the performance metrics are computed as an average of ten independent learning instances each originating from a distinct random seed, with standard deviation bands computed where appropriate. Additionally, for agents sharing the same architecture (i.e. regularized and not regularized agents) learning is conducted in pairs with an identical set of starting parameters for each, so as to provide an apples-to-apples comparison. Fig. 2.6b depicts the comparison of agent performance using the relative amount collected metric $V(e, P_{200})$. At first glance, the performance of DQN and monotonicity regularized DQN is almost identical, while the collection performance of the

spline approximator is noticeably lagging behind, especially in the initial training stages. A plausible determining factor is the vast number of parameters (flexibility) the neural network wields (8,902 individual weights) over the B-spline approximator (252 individual weights). On the other hand, by the episode 19,000 the linear B-Spline approximator manages to catch up with the performance of the vastly more complex neural network fitted DQN. Hence, from a performance per parameter perspective, a linear B-Spline approximator is favored over a black box neural network.

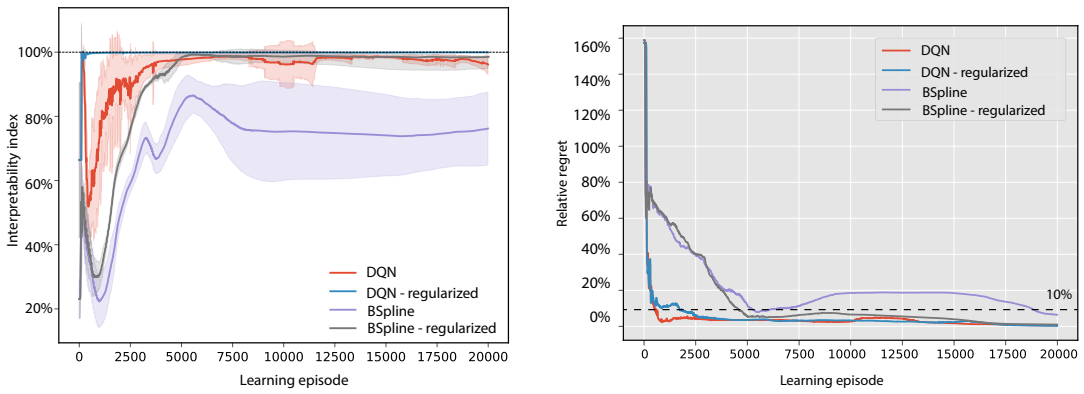
Additionally, all agents relatively quickly discover a superior policy to the autonomous collection (i.e., always select inaction), and except for the non-regularized B-Spline learner, all finish with a policy that is performance-wise comparable to the theoretically unattainable optimum. The speed of convergence can be judged from the regret metric $G(e, P_{200})$. Fig. 2.7b demonstrates that both DQN variants feature comparable convergence speeds while the regularized B-Spline agent needs twice the time to reach a 10% regret mark, and the non-regularized B-Spline agent requires more than six times the training time of the vanilla DQN (2,500 vs 5,000 vs 18,000 episodes).

Finally, the interpretability index that counts the number of violations of monotonicity constraints as a percentage of the entire statespace non-surprisingly favors the regularized learning agents (Fig. 2.7). The DQN monotonicity regularized learner produces sensible value functions and policies already from the first couple hundred learning episodes. In contrast, DQN that relies on picking up monotonicity from the data samples takes 4000 learning steps longer while still exhibiting variance across the different learning instances.

To reiterate, in contrast to the work by Chehrazi et al. (2019) the policies derived using the learning agent in Section 2.3 do not have an explicit access to the exact parameter values. The repayment-process specification is reflected only in the stochastic feedback the collector receives in the form of repayments. This carries a significant advantage over an analytically derived policy, as the learning agent is immediately redeployable in other collection environments with arbitrarily modified dynamics; i.e., one can *hot swap* the environment in Fig. 2.2, and the agent is still capable of learning "high-quality" policies—merely from the (possibly synthetic) data stream of repayment events.[8] As a matter of fact, a more realistic model of repayment behavior would likely exhibit a non-stationary, state-dependent relative repayment distribution. Although straightforward to implement, extensions such as this one more often than not yield analytically intractable problems;[9] however, our solution can be out-of-the-box plugged in as the only requirement is to be able to sample the repayment events.

[8] For realistic environments, the data stream is usually synthetic based on identified repayment-process parameters (Mark and Weber, 2020), thus reducing the learning time to the simulation runtime rather than real collection time.

[9] Even if an analytical solution might eventually be obtained for a given environment, its derivation would present important time delays relative to the more immediate collection tasks at hand.



(a) (b)

Figure 2.7: (a) Evolution of the interpretability index I. (b) Evolution of the learning regret $G(e, P_{200})$.

κ	λ_0	λ_∞	δ_{10}	δ_{11}	δ_{20}	δ_{21}	ρ	c
0.7	0.11	0.1	0.02	0.5	0.0	1.0	15%	\$10

Table 2.1: Specification of the reference repayment process for the numerical experiment.

2.5 Conclusion

In this paper, we combined a number of state-of-the-art reinforcement-learning results to develop a learning agent for the practice of credit collections. Firstly, in contrast to the majority of contemporary reinforcement-learning applications, the collections problem features an asynchronous action-feedback relationship. That is, a reward for an action taken at state s is observed at some later time based on the intricate dynamics of the system. Even though this asynchronous feature is characteristic for a vast number of practically relevant problems (such as human learning, optimal execution, etc.), the literature on this topic is sparse, with the vast majority of reinforcement-learning contributions focusing on synchronous action-feedback environments. To this end, we formulate a stochastic reward shaping theorem which is applicable to all asynchronous environments driven with Hawkes-like state-space dynamics. This result straightforwardly transforms otherwise discretely observed reward into its continuous analogue, and thus allows learning in these challenging environments. Secondly, with respect to the growing need for interpretable and *ethical* machine-learning models, a regularization technique that produces consistent and interpretable policies is introduced. In essence, our regularizer naturaly incorporates structural insights in form of monotonicity and/or convexity constraints, and thus enables learning of interpretable value functions and policies. Our experiments demonstrate that the monotonicity penalization term does not affect the overall performance of the learned policies or their speed of convergence; however, it guarantees interpretability and consistency of the learned results via the imposed structural constraints. The importance of interpretable reinforcement learning spans well beyond the discussed

application to credit collections, and is one of the principal causes hindering wide adoption of reinforcement learning based decision making in the business context. In contrast to a theoretical lab experiment, practical applications require learned policies to be subjected to a human decision maker's oversight for validation. Thus, in practical settings, decision maker values consistency, interpretability, and understandability even at the price of slightly suboptimal performance, as these are key components for auditability of the model. This in turns allows decision maker to explain the policy to a third party (including a benevolent court of law if necessary) and can therefore provide a clear rationale (be it *ex ante* or *ex post*) for the implementation of machine-learned actions. In our setting of a stochastic control problem with asynchronous rewards we have shown that interpretability regularization in fact guides the learning agent to fully interpretable policies. To quantify the generic suitability of a learned policy, we have proposed an interpretability index (percentage of monotonicity-constraint adherence of the learned value function) which clearly demonstrates the benefit of monotonicity regularized q-learning. In this way, the paper contributes to the broader discussion on ethical machine learning and its implications for business applications. The newly developed agents were tested against a suite of two other benchmark policies – an autonomous policy and an optimal policy for an analytically traceable version of the collection problem. All the agents demonstrated a consistently superior performance to the autonomous inaction policy, while the DQN based agents delivered performance only 5% below the theoretically unattainable optimal solution. The significance of our agent lies in its ability to learn from mere interactions with the environment (debtor), without knowing any analytical details about the repayment process. This makes our agent highly applicable to other repayment processes (including fully data-driven environments), and invites future research with more complex (and more realistic!) model dynamics.

3 Domain-Knowledge Enhanced Policy Gradient: Application to Credit Collections

This chapter is based on Mark, M., Chehrazi, N., Liu, H., and Weber, T. A. (2021). Optimal Recovery of Unsecured Debt via Interpretable Reinforcement Learning. Working Paper, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

3.1 Introduction

Reinforcement learning has become a popular computational approach for solving real-life sequential decision-making problems. Over the past few years, it has been steadily gaining momentum, especially because of its success in complex high-dimensional control tasks such as playing the Atari game suite (Mnih et al., 2013) or Starcraft at super-human levels (Vinyals et al., 2019). Despite such celebrated breakthroughs, reinforcement learning has not yet been broadly adopted by businesses for solving more traditional operations research (OR) problems. This is often attributed to the data-hungry nature of these algorithms, which makes them suitable only in applications where large amounts of data can be generated on demand (e.g., in robotics). Furthermore, business applications tend to impose additional requirements on machine learning (ML) models that go well beyond mere performance goals, such as the *interpretability* of the resulting decision rules and thus their comprehensibility for human decision makers. For instance, when deciding on how much credit to extend to a car-loan applicant, we expect this point estimate to be not only sufficiently accurate, but also monotonically increasing in the applicant's salary and credit rating. However, when training a neural network or any other highly flexible approximator on real data, we risk to locally overfit and thereby obscure this intuitive and important relationship. Consequently, the local inconsistencies in this dependency produced by standard ML methods would tend to undermine a decision maker's confidence in the decision rule, and as a result such a model would not stand a good chance of getting implemented—despite possibly a good numerical performance overall. Should the model nevertheless pass the validation phase and be adopted in practice, it is prone to produce locally biased predictions, which would predominantly affect underrepresented subgroups (e.g., minorities) for which the available data are relatively

sparse. Therefore, the notion of interpretability and systemic consistency is closely tied to the broader challenge of *ethical* machine learning (Piano, 2020).

In practice, the challenge of interpretable (ethical) ML—often tied to monotonicity and/or convexity constraints of the learned policy with respect to its inputs—has been steadily gaining attention in the literature. For instance, You et al. (2017) propose a deep lattice framework (as a counterpart to neural nets) to learn flexible monotonic functions, and Gupta et al. (2019) regularize the element-wise loss with local monotonicity constraints to encourage learning of monotonic neural nets. Similarly, when developing a decision-making system based on reinforcement learning for business use cases, we require that learned policies be not only performant but also intuitive and understandable, whence interpretable by human decision makers. One possible way for achieving policy interpretability is by embedding structural knowledge into the learner itself. Many practically relevant problems benefit from an extensive theoretical analysis of the properties of their value functions and optimal policies. However, this structural knowledge is usually discarded in an ML setting, for a lack of systematic procedure for incorporating structural domain knowledge. In this paper, we propose an adapted deep deterministic policy gradient method that incorporates expert domain knowledge directly into the learning process to obtain interpretable policies. For this we introduce a monotonicity regularizer for the actor's loss function which penalizes deviations of policies from structural properties during the learning procedure. Intuitively, this regularization filters out undesirable local minima in the policy space by means of an augmented loss gradient that pushes solutions away from non-interpretable regions towards complete interpretability, at comparable performance. As a result, we achieve more stable learning with less variance across runs. We showcase the relevance of our approach in the context of optimal credit collections, a practically relevant stochastic control problem which features a self-exciting (Hawkes) repayment process and an asynchronous learning feedback.

3.2 Background

3.2.1 Preliminaries

We study a specific type of reinforcement learning problems, the solution to which may benefit significantly from structural input provided by domain experts. This is often the case for control problems in OR, finance, or economics. Specifically, our method is illustrated by a problem of optimal credit collections which bridges these three areas. The results can be readily applied to other problems where structural knowledge can be cast in terms of monotonicity constraints.

The collection problem is an OR problem broached by Mitchner and Peterson (1957), often aptly compared with the game of poker. The collector observes a stochastic sequence of marked temporal repayment events $(\tau_i, b_i)_{i\geq 1}$, where τ_i and b_i denote the i-th repayment time and repayment magnitude, respectively. To maximize the present value of the revenue

stream, the collector has the option to perform costly collection actions, a_t at time $t \geq 0$, that temporarily increase the likelihood of repayment events. Just as in poker, committing to actions (betting) takes place before the full collection (completion of hand) is observed. Thus, to stay in the game betting must continue.

We specify the collections problem as a Markov decision process (MDP) with a state space $\mathscr{S}$, an action space $\mathscr{A}$, transition probabilities $P(s_{k+1}, s_k, a_k)$, an initial state distribution ρ_0 (on $\mathscr{S}$), a reward function $\mathscr{R} : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \to \mathbb{R}$, and a discount factor $\gamma \in (0,1)$. The MDP is a discrete-time counterpart of the continuous-time repayment process introduced by Chehrazi and Weber (2015) in terms of a stochastic differential equation (SDE),

$$d\lambda(t) = \underbrace{\kappa(\lambda_\infty - \lambda(t))dt}_{\text{mean-reversion}} + \underbrace{\delta_1^\top dJ(t)}_{\text{self-excitation}} + \underbrace{dA(t)}_{\text{collection strategy}}, \quad t \geq 0. \tag{3.1}$$

This mean-reverting SDE describes the intensity dynamics of an account placed in collections at time $t = 0$ with (given) initial intensity $\lambda(0) = \lambda$. Eq. (3.1) can be derived from a continuous-time hidden Markov process where an account holder can be in one of two distinct states, "H" or "L." A representative account holder in state "H" would make random partial repayments at higher frequency than if he was in state "L." The account holder's state evolves according to a generic Markov jump process which can be positively influenced by the credit-issuer through costly collection actions. While the state cannot be observed directly by the collector, he can estimate the likelihood of the account holder's being in either state "H" or "L"—based on the observed repayment history. The Bayesian dynamics of these estimates translate to the SDE specification in Eq. (3.1). In particular, the self-excitation term captures a discrete upward adjustment in the collector's beliefs upon observing a repayment. The jump is positive, since a repayment is more likely in state "H" than in state "L." In that description of the intensity dynamics, the vector $J(t) = [N(t), Z(t)]^\top$ consists of an unmarked counting process $N(t) = \sum_i \mathbb{1}\{\tau_i \leq t\}$ and its marked counterpart $Z(t) = \sum_i z_i \mathbb{1}_{\{\tau_i \leq t\}}$. The marks represent relative repayments, drawn from an empirically identifiable distribution F_z on a support in $[z_{\min}, 1]$, with a positive minimum $z_{\min}$. The vector $\delta_1^\top = [\delta_{10}, \delta_{11}]$ describes the sensitivity of the process to repayment events. In the absence of a repayment, the effective rate of repayment $\lambda(t)$ declines, since a period of inactivity is more likely in state "L" than state "H." This is captured by the first term in Eq. (3.1), where the parameter λ_∞ denotes the steady-state of the effective repayment intensity and κ the rate of convergence. The latter parameter, which shapes the covariance properties of the process, determines how much "memory" the system retains. Unlike the intensity dynamics in Eq. (3.1) (for $\lambda(t)$), the dynamics of the outstanding balance $w(t)$ are relatively simple: At any repayment time τ_i, the account's outstanding balance $w(\tau_i)$ diminishes by the amount b_i repaid, i.e., $w(\tau_i) = (1 - z_i)\, w(\tau_i^-)$, where $z_i = b_i / w_{i-1}$ for $i \geq 1$. Hence,

$$w(t) = w(\tau_i), \qquad \tau_i \leq t < \tau_{i+1}. \tag{3.2}$$

Lastly, in the absence of a collection strategy $A(t)$, the Markovian nature of the process allows

for a compact representation,

$$\lambda\big(t'|\lambda(t)\big) = \varphi\big(t', \lambda(t)\big) = \lambda_\infty + (\lambda(t) - \lambda_\infty)e^{-\kappa t'}, \quad t' \geq t, \tag{3.3}$$

which describes the law of motion for the intensity starting at $\lambda(t)$, provided no repayments were received on the interval $[t, t']$.

To cast the collections problem into a reinforcement-learning framework, the continuous-time Markovian dynamics in Eqs. (3.1) and (3.2) must be expressed as a discrete-time Markov chain. In particular, measuring time in small discrete steps of Δt, we assume—without loss of generality—that actions are taken at the beginning of an interval $[k\Delta t, (k+1)\Delta t]$ while repayments, if they occur, are received at the end of such an interval. In fact, this assumption is required to make the discrete-time repayment process non-predictable. From the Poisson dynamics of the repayment process, the likelihood of receiving a repayment at the end of the interval $[k\Delta t, (k+1)\Delta t]$, given initial intensity $\lambda(k\Delta t)$ and action $a_{k\Delta t}$, is

$$\mathbb{P}[N\big((k+1)\Delta t\big) - N(k\Delta t) = n|\mathscr{H}_{k\Delta t}] = \begin{cases} 1 - \big(\lambda(k\Delta t) + a_{k\Delta t}\big)\Delta t + o((\Delta t)^2), & n = 0, \\ \big(\lambda(k\Delta t) + a_{k\Delta t}\big)\Delta t + o((\Delta t)^2), & n = 1, \\ o((\Delta t)^n), & n \geq 2. \end{cases} \tag{3.4}$$

In the previous equation,[1] the discrete-time dynamics of $\lambda(k\Delta t)$ for $k \in \mathbb{Z}_+$ follow:

$$\lambda(k\Delta t) = \varphi\big(\Delta t, \lambda\big((k-1)\Delta t\big) + a_{(k-1)\Delta t}\big) + (\delta_{10} + \delta_{11} z_{k-1})\mathbb{1}_{\{N(k\Delta t) - N((k-1)\Delta t) \neq 0\}}, \tag{3.5}$$

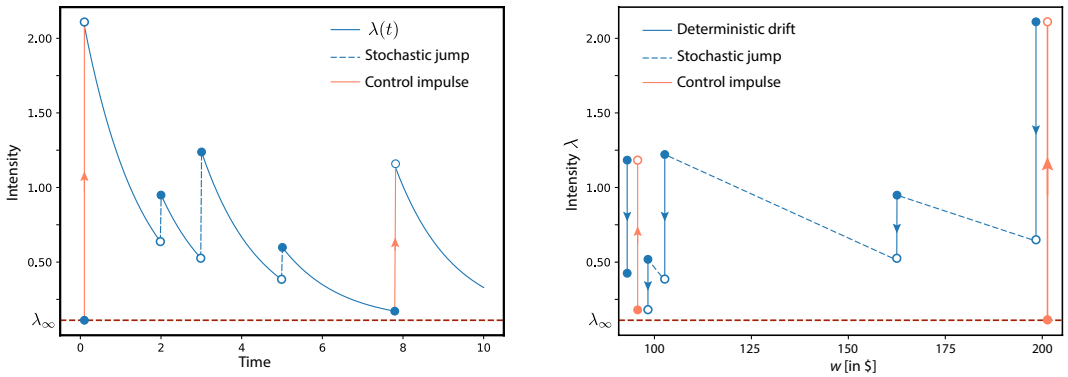
with $\lambda(0) = \lambda_0$, where we are allowed to use Eq. (3.3), since no discrete event will take place on the interval $(k\Delta t, (k+1)\Delta t)$. Finally, the z_k are independent and identically distributed (i.i.d.) draws from the relative-repayment distribution F_z, so the account balance evolves according to

$$w(k\Delta t) = (1 - z_{k-1})\, w\big((k-1)\Delta t\big)\mathbb{1}_{\{N(k\Delta t) - N((k-1)\Delta t) \neq 0\}}, \quad k \geq 0, \tag{3.6}$$

with $w(0) = w_0$ and $z_{-1} = 0$. Equations (3.4)–(3.6) fully describe the discrete-time dynamics of the collection process. To simplify the notation, in what follows we denote the tuple $\big(\lambda(k\Delta t), w(k\Delta t), a_{k\Delta t}\big)$ by (λ_k, w_k, a_k). In our numerical implementation, the value of (λ_k, w_k) is discretized on the set of reachable states (λ, w), denoted by $\mathscr{S} \subsetneq \mathbb{R}^2_+$. This last step turns the discrete-time, continuous-space Markov dynamics of Eq. (3.4)–(3.6) to a discrete-time finite Markov chain, but otherwise this computational simplification is not critical for our theoretical developments. It is important to note that we do *not* restrict attention to the discrete grid of states, but rather use it to partition the state-space exploration. The repayment-process dynamics are illustrated in Fig. 3.1.

We can now consider the discrete state-space dynamics, introduced above, as our reinforcement-learning setting. In particular, consider the behavior of the two parties involved: a *decision maker* (also referred to as *agent*) and an *environment* that is responsible for providing feed-

[1] $\mathscr{H}_t$ is the information filtration generated by observable events up to time t.



a) b)

Figure 3.1: Controlled state-transition dynamics with two action-induced jumps. a) Self-exciting intensity $(t, \lambda(t))$. b) Account state $(\lambda, w) \in \mathscr{S}$.

back on the agent's action in terms of some *reward*.[2] The environment behavior is described by Eqs. (3.4)–(3.6). The agent, following a policy $\pi : \mathscr{S} \to \mathbb{R}_+$ that prescribes his action for a given state, repeatedly interacts with the environment. At each (discretized) time step $k \geq 0$, the agent observes his state $s_k = (\lambda_k, w_k) \in \mathscr{S}$, selects an action $\pi(s_k) = a_k \in \mathscr{A} = \mathbb{R}_+$ according to policy π, and the environment responds (stochastically) with the subsequent state $s_{k+1} = (\lambda_{k+1}, w_{k+1})$, together with a random reward $r_k \in \mathbb{R}$ associated with the state transition from s_k to s_{k+1} which is of the form

$$r_k \triangleq \mathscr{R}(s_k, a_k, s_{k+1}) = \begin{cases} \gamma z_k w_k - c a_k, & \text{repayment received in } [k\Delta t, (k+1)\Delta t], \\ -c a_k, & \text{no repayment in } [k\Delta t, (k+1)\Delta t], \end{cases} \tag{3.7}$$

where $\gamma = \exp(-\rho\Delta t)$. The agent's goal is to find a policy π that maximizes net collections,

$$v_\pi(s_0) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^{k+1} w_k z_k - c \sum_{k=0}^{\infty} \gamma^k a_k \middle| \mathscr{H}_0 \right], \tag{3.8}$$

with $a_k = \pi(s_k)$ and a given initial state $s_0 = (\lambda_0, w_0)$.

3.2.2 Deterministic Policy Gradient Theorem

The Deterministic Policy Gradient (DPG) is a policy gradient method suitable for control tasks with continuous action spaces (Silver et al., 2014). In contrast to the standard *stochastic* policy gradient, DPG aims to learn a *deterministic* policy $\pi_{\boldsymbol{\theta}} : \mathscr{S} \to \mathscr{A}$ with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{d_1}$ of dimension $d_1 \ll |\mathscr{S}|$. Let $\rho_{\pi_\theta}(s') = \int_{\mathscr{S}} \sum_{t=0}^{\infty} \gamma^t \rho_0(s) P_t(s, s'; \pi_{\boldsymbol{\theta}}) ds$ be the discounted state-visitation distribution, where $P_t(s, s'; \pi_{\boldsymbol{\theta}})$ denotes the probability of going from s to s'

[2] In engineering applications, the terms *system, controller* and *control signal* are used synonymously for the terms *environment, agent,* and *action* employed here.

in t steps under a policy $\pi_{\boldsymbol{\theta}}$, i.e., $\mathbb{P}(s_{k+t} = s' | s_k = s, \pi_{\boldsymbol{\theta}})$.[3] We define an optimal policy $\pi^{\star}_{\boldsymbol{\theta}}$ such that $\pi^{\star}_{\boldsymbol{\theta}} \in \arg\max_{\theta} J(\pi_{\boldsymbol{\theta}})$, where

$$J(\pi_{\boldsymbol{\theta}}) \triangleq \mathbb{E}_{s_0 \sim \rho_0}[v_{\pi_\theta}(s_0)] = \int_{\mathscr{S}} \rho_{\pi_\theta}(s) r(s, \pi_{\boldsymbol{\theta}}(s)) ds = \mathbb{E}_{s \sim \rho_{\pi_\theta}}[r(s, \pi_{\boldsymbol{\theta}}(s))] \tag{3.9}$$

where $r(s, \pi_{\boldsymbol{\theta}}(s)) = \mathbb{E}_{s' \sim P_1(s,s';\pi_\theta)}[\mathscr{R}(s, \pi_{\boldsymbol{\theta}}(s), s')]$. By the deterministic policy gradient theorem of Silver et al. (2014), we have

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \int_{\mathscr{S}} \rho_{\pi_\theta}(s) \nabla_a q_{\pi_\theta}(s,a)|_{a=\pi_{\boldsymbol{\theta}}(s)} \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(s) ds = \mathbb{E}_{s \sim \rho_{\pi_\theta}}\left[\nabla_a q_{\pi_\theta}(s,a) \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(s)|_{a=\pi_{\boldsymbol{\theta}}(s)}\right], \tag{3.10}$$

where $q_{\pi_\theta}(s,a) = r(s,a) + \gamma \int_{\mathscr{S}} P_1(s', s, \pi_{\boldsymbol{\theta}}) v_{\pi_\theta}(s') ds'$ is the q-function associated with Eq. (3.8) for policy π_θ. A number of extension algorithms were derived from the vanilla DPG, arguably the most popular one being Deep DPG (DDPG) (Lillicrap et al., 2015), an off-policy actor-critic type algorithm that combines DPG and double q-learning (Hessel et al., 2017). In this setting, the q-function (critic) is parametrized with parameters $\mathbf{w} \in \mathbb{R}^{d_2}$, i.e., $q_{\pi_\theta}(s,a) = \hat{q}_{\pi_\theta}(s,a;\mathbf{w})$, and is learned by a minimizing loss sequence, of the form

$$\mathscr{L}(\mathbf{w}_l) = \mathbb{E}_{(s_k, a_k, r_k, s_{k+1}) \sim D}\left[\frac{1}{2}\Big((r_t + \gamma \hat{q}_{\pi_\theta}(s_{k+1}, \pi(s_{k+1}); \mathbf{w}_{l-1}) - \hat{q}_{\pi_\theta}(s_k, a_k; \mathbf{w}_l)\Big)^2\right], \tag{3.11}$$

for $l \geq 1$, where the distribution D samples from a memory buffer of *uncorrelated* experience samples (Fedus et al., 2020), and $\mathbf{w}_{l-1}$ is a vector of previously estimated parameters—with $\mathbf{w}_0$ being randomly initialized at the start of training. The actor $\pi_{\boldsymbol{\theta}}(\cdot)$ then ascends in the direction of the gradient of the objective function

$$J^{(l)}_{\beta}(\pi_{\boldsymbol{\theta}}) = \int_{\mathscr{S}} \rho_\beta(s) v_{\pi_\theta}(s) ds = \mathbb{E}_{s \sim \rho_{\beta(\cdot)}}\left[\hat{q}_{\pi_\theta}(s, \pi_{\boldsymbol{\theta}}(s); \mathbf{w}_l)\right], \tag{3.12}$$

where $\beta : \mathscr{S} \times \mathscr{A} \to [0,1]$ is an arbitrary, possibly stochastic, exploration distribution (*behavioral* policy) such that $\int_{\mathscr{A}} \beta(s,a) da = 1$ for all $s \in \mathscr{S}$. The gradient of this modified objective can still be easily computed, yet the off-policy training implied by the flexible use of β provides a better stability and sample efficiency. Furthermore, policy gradient algorithms typically require some sort of importance sampling for both actor and critic that reweighs the rewards so as to reflect that actions were taken according to β rather than π. However, because DPG uses temporal-difference updates for the critic and the policy is deterministic (i.e., the integral over the actions in the objective function disappears), we can avoid importance sampling altogether.

3.3 Incorporating Domain Knowledge

For the optimal collection problem, the interpretability of a given policy π to a human collector is closely linked to the structure of the action set in the state space. This structure must

[3] By abuse of notation (and terminology), ρ_π is an *improper* distribution, so generically: $\int_{\mathscr{S}} \rho_\pi(s) ds \neq 1$.

follow systemic consistency conditions which can be framed in terms of policy monotonicity: first, actions for a fixed account balance w cannot increase when the repayment intensity λ increases; and second, the actions cannot decrease in the account balance w when the repayment intensity λ is held constant. That is,

$$\big(w' \leq w \;\; \Rightarrow \;\; \pi_{\boldsymbol{\theta}}(\lambda, w') \leq \pi_{\boldsymbol{\theta}}(\lambda, w)\big) \;\; \textit{and} \;\; \big(\lambda \leq \lambda' \;\; \Rightarrow \;\; \pi_{\boldsymbol{\theta}}(\lambda', w) \leq \pi_{\boldsymbol{\theta}}(\lambda, w)\big). \tag{3.13}$$

These consistency conditions, which impose shape constraints on the policy, capture the economic logic that if it is optimal to act for an account in a lower balance state, then it must also be optimal to act (at least as forcefully) for an account at a higher balance, and similarly, an account in lower intensity state is less likely to repay, so an optimal action has to be at least of the same size. For a detailed analysis of the theoretical properties of policy and value function, see Chehrazi et al. (2019) who obtain an optimal solution for the collection problem in continuous time. The monotonicity constraints in Eq. (3.13) can be included in the learning by means of a barrier regularization term,

$$H(\pi_{\boldsymbol{\theta}}(\lambda, w)) = \eta_1 \max\{0, \frac{\partial \pi_{\boldsymbol{\theta}}(\lambda, w)}{\partial \lambda}\} + \eta_2 \max\{0, -\frac{\partial \pi_{\boldsymbol{\theta}}(\lambda, w)}{\partial w}\}, \tag{3.14}$$

where η_1 and η_2 are (generally distinct) penalization constants. Similar to a maximum-entropy policy gradient framework where a regularizer encourages learning of explorative policies (see, e.g., Haarnoja et al. (2018)), we add the regularizer to the off-policy performance metric in Eq. (3.12), so

$$\hat{J}(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{s\sim\rho_{\beta}(\cdot)}\left[q_{\pi_\theta}(s, \pi_{\boldsymbol{\theta}}(s)) - H(\pi_{\boldsymbol{\theta}}(s))\right]; \tag{3.15}$$

this "domain-knowledge enhanced objective" still allows for a straightforward computation of the gradient, as

$$\nabla_{\boldsymbol{\theta}} \hat{J}(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{s\sim\rho_{\beta}(\cdot)}\left[\nabla_a \hat{q}_{\pi_\theta}(s, a)|_{a=\pi_{\boldsymbol{\theta}}(s)} \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(s) - \nabla_\theta H(\pi_{\boldsymbol{\theta}}(s))\right]. \tag{3.16}$$

The intuition behind the shape regularizer (which can easily be augmented to also contain higher-order monotonicities, for example to capture the concavity of the action frontier with respect to w) is to reject critical points in the policy space that yield locally uninterpretable policies (i.e., violating Eq. (3.13)) in favor of parameters satisfying the systemic consistency constraint while staying within an ϵ-neighborhood in the parameter space. For a full learning algorithm of the domain-knowledge enhanced DPG (DKEDPG), see Alg. 3 specified hereafter.

3.4 Results

Our numerical study comprises 50 independent runs of DDPG (non-penalized) and DKEDPG (penalized) algorithms on the collection problem, each for 10,000 episodes. An episode consists of a full collection trajectory from the initial account state s_0 to its final state s_T at the end of the time horizon T, where s_0 is randomly initialized as a uniformly distributed draw

Algorithm 3: Domain-Knowledge Enhanced Deterministic Policy Gradient (DKEDPG).

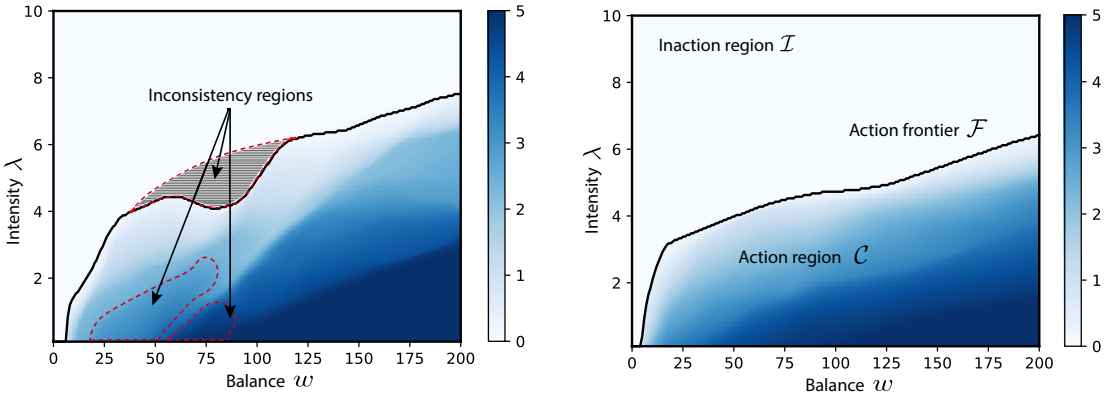
Algorithm parameters:
$(\lambda_0, \lambda_\infty, \kappa, \delta_1, \delta_2)$ - process parameters, Δt- discretization step, N_{episodes}- number of episodes, ζ- exploration noise, $\xi \in (0,1)$- update sensitivity coefficent, L - batch size

Randomly initialize critic network $\hat{q}_{\pi_\theta}$ and actor network $\pi_{\boldsymbol{\theta}}$ with parameters $\mathbf{w}$ and $\boldsymbol{\theta}$
Initialize target network $\hat{q}'_{\pi_\theta}$ and $\pi'_{\boldsymbol{\theta}}$ with weights $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$ and $\mathbf{w}' \leftarrow \mathbf{w}$
Initialize the replay buffer D [state-transition history with uniform sampling]
for *episode=1:*$N_{episodes}$ **do**
  Select a starting state $s_0 = (\lambda_0, w_0)$ according to $\rho_0(\cdot)$
  Set $k = 0$ **while** s_k *is non-terminal (i.e.,* $w_k \geq 1$*)* **do**
    Select action $a_k = \pi_{\boldsymbol{\theta}}(s_k) + \zeta$ according to the current policy and exploration noise
    Take an action a_k, observe reward r_k, next state s_{k+1}, and a Boolean flag indicating whether s_{k+1} is terminal state or not
    Store the transition (s_k, a_k, r_k, s_{k+1}) in the experience replay D
    Sample a random minibatch of transitions $B = \{(s_l, a_l, r_l, s_{l+1})\}_{l=1}^{L}$ according to D
    Set $y_l = \begin{cases} r_l, & \text{for terminal } s_{l+1}, \\ r_l + \gamma \hat{q}'_{\pi_\theta}(s_{l+1}, \pi'_{\boldsymbol{\theta}}(s_{l+1}); \mathbf{w}') & \text{for non-terminal } s_{l+1}. \end{cases}$
    Update the critic weights $\mathbf{w} \in \arg\min_{\mathbf{w}} \frac{1}{\|B\|} \sum_{l=1}^{L} \left[\left(y_l - \hat{q}_{\pi_\theta}(s_l, a_l; \mathbf{w}) \right)^2 \right]$
    Compute the constraint-violation penalty $H(\pi_{\boldsymbol{\theta}}(s_l))$
    Update the actor policy using sampled policy gradient:
    $\nabla_{\boldsymbol{\theta}} \hat{J}(\pi_{\boldsymbol{\theta}}) = \frac{1}{\|B\|} \sum_{l=1}^{L} \left[\nabla_a \hat{q}_{\pi_\theta}(s_l, a)_{|a=\pi_{\boldsymbol{\theta}}(s_l)} \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(s_l) - \nabla_\theta H(\pi_{\boldsymbol{\theta}}(s_l)) \right]$
    Update the target networks:
    $\boldsymbol{\theta}' \leftarrow \xi \boldsymbol{\theta} + (1-\xi)\boldsymbol{\theta}'$
    $\mathbf{w}' \leftarrow \xi \mathbf{w} + (1-\xi)\mathbf{w}'$
  end
end

from the state space $\mathscr{S}$.[4] Importantly, in order to not stall learning in early stages, we turn the monotonicity regularization on from episode 800 onwards (until then the penalization coefficients are set to zero). To isolate the exact effect of the interpretability regularizer H on learning, every pair of DDPG and DKEDPG runs is seeded with an identical randomization seed and initialized using the same network weights.

Fig. 3.3a displays the performance evolution of both agents, averaged over all 50 runs (analogous to learning curves). Since their performance is quasi-identical, we observe no performance-related cost from implementing policy regularization. In particular, the two learning agents' performance is identical during the first 800 episodes, due to the same randomization seed and initial network weights, and it starts to differ only once the policy regularization is activated.

[4] For the implementation details and specific parameters, see Appendix B.1.



a) b)

Figure 3.2: Interpretability of a state-control feedback policy. a) Monotonicity violation. b) Fully interpretable policy.

In Sec. 3.3, we provide a link between interpretability and policy monotonicity in the state-space. Fig. 3.2 demonstrates the intuitive meaning of interpretability. The shaded regions represent the action set $\mathscr{C}$ where the collector exerts positive intensity impulses with magnitude illustrated by the heat map. Arguably the most important feature of the policy is its action frontier $\mathscr{F}$, i.e., the interface between $\mathscr{C}$ and inaction region $\mathscr{I}$. The salient systemic inconsistency of the non-penalized policy is exhibited by the nonmonotonic and non-concave shape of the action frontier (resulting in a non-convex action set). Accordingly, under such an inconsistent policy any accounts in states s outside the closure of $\mathscr{C}$ but still in the (closed) convex hull of $\mathscr{C}$, would be discriminated against in treatments. Furthermore, given the required policy monotonicity in Eq. (3.13), with increasing balance (resp., intensity) we expect gradually increasing (resp., decreasing) magnitudes of the actions (i.e., no islands in the heat map), a feature clearly violated by the non-penalized agent in Fig. 3.3. Policy interpretability is assessed using two distinct metrics. First, we define an *interpretability index* (with respect to the policy monotonicity required in the application) as

$$I = \frac{1}{\|\mathscr{C}\|}\int_{\mathscr{C}} \mathbb{1}_{\left\{\left(\partial_\lambda \pi_{\boldsymbol{\theta}}(\lambda,w)\leq\delta\right)\wedge\left(-\delta\leq\partial_w \pi_{\boldsymbol{\theta}}(\lambda,w)\right)\right\}} ds, \tag{3.17}$$

where $\delta > 0$ denotes some tolerance for non-monotonicity (zero being the most strict), and $ds = d\lambda \times dw$ denotes the standard (Lebesgue-)measure on $\mathscr{S}$. The monotonicity measure can be interpreted as a relative number of non-violations in the action set $\mathscr{C}$, i.e., how many percent of the action set is interpretable. Fig. 3.3b depicts the time evolution of the non-violation (compliance) metric in number of episodes averaged over all runs. The penalization clearly brings the desired effect producing interpretable policies almost immediately while non-penalized DPG attains only 90% interpretable policies at the learning termination with far greater variance among the runs.

Second, we introduce a *systemic consistency index* (C_ℓ) (again, with respect to policy mono-

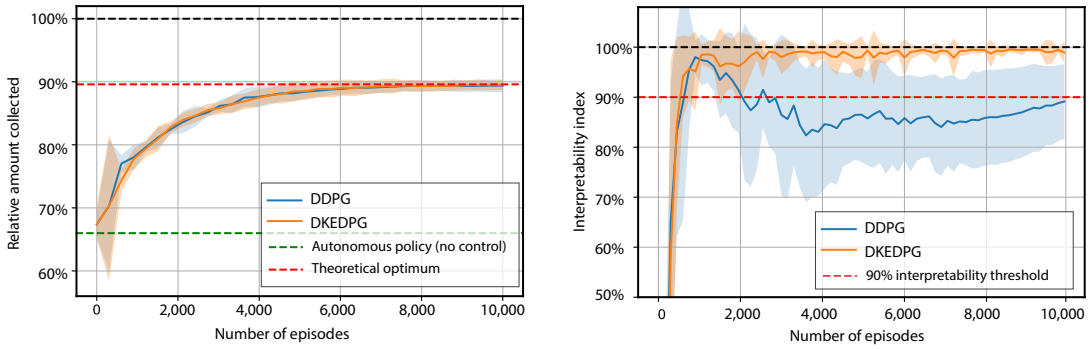
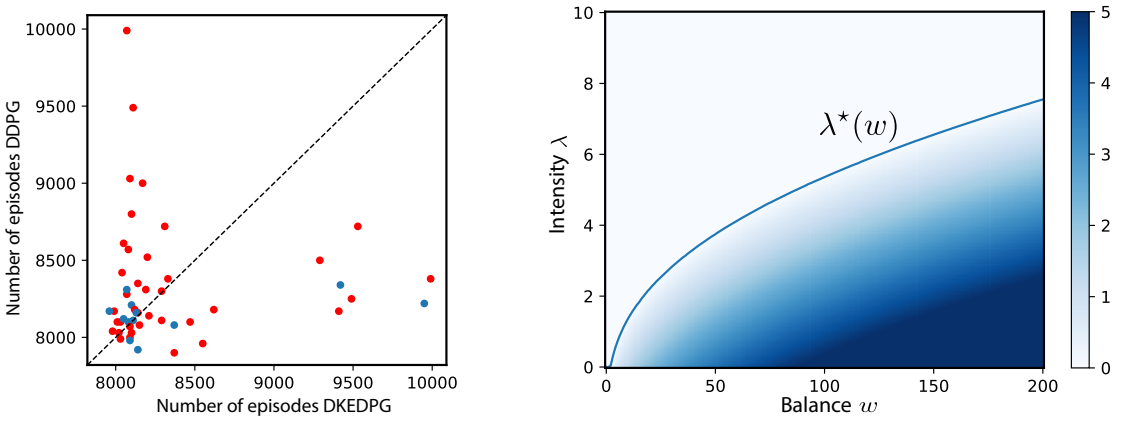


Figure 3.3: Comparison of learned policies under DKEDPG and standard DDPG. a) Net collections. b)Interpretability index

tonicity) so as to connect interpretability to the agent's learning performance. For this we consider a learning stopping step K^α such that for all $k \geq K^\alpha$ (within a sufficiently large learning horizon T) the norm of the gradient does not exceed a given positive threshold α, i.e., $\|\nabla_\theta J(\pi^k_{\boldsymbol{\theta}})\| \leq \alpha$. This simple (yet effective) stopping rule uses the fact that the norm of the learning gradient vanishes approximately near critical points in the policy space. Given this stopping criterion, to determine C we measure both the stopping time and the interpretability index at the stopping episode $k = K^\alpha$. Fig. 3.4a depicts this relationship on a comparison graph in the spirit of the well-known "q-q plot." We observe that both agents perform similarly in terms of convergence, with a majority of points being uniformly dispersed around the 45-degree line. However, as for interpretability only 11 out of the 50 non-penalized runs terminated with an interpretable policy at the $\ell = 95\%$ level of violations (corresponding to $I = 95\%$), so $C_{95\%} = 11/50 = 22\%$. For $\ell = 99\%$ the systemic consistency of the non-penalized agent drops to zero. By contrast, the penalized agent terminated with an interpretable policy at the 99% level in all 50 runs, thus attaining perfect systemic consistency at $C_{99\%} = C_{95\%} = 100\%$. This indicates that incorporating the interpretability regularizer rendered all policies interpretable, without any noticeable loss in average performance or convergence speed.

Comparison with theoretical optimum. To highlight and address deficiencies of data-learned policies, we purposefully selected an analytically well-explored practical problem. Indeed, Chehrazi et al. (2019) derive an optimal solution for the collection problem with a value function in semi-closed form (see Fig. 3.4b for the corresponding optimal state-feedback control law). However, despite knowing the theoretical optimum in this particular setting, the reinforcement learning approach goes one step further by easily carrying over to analytically intractable variants of the problem (e.g., with state-dependent repayment distributions or actions with memory). Finally, given a theoretical solution in our setting it is possible to compare the performance of both agents against this exact benchmark. From a perspective of accounts outside of the action region the only relevant part of the policy is the action frontier. Therefore, in Fig. 3.5a we measure mean squared error (MSE) of both agent-learned



a) b)

Figure 3.4: a) Interpretability of non-penalized agent. b) Theoretically optimal policy.

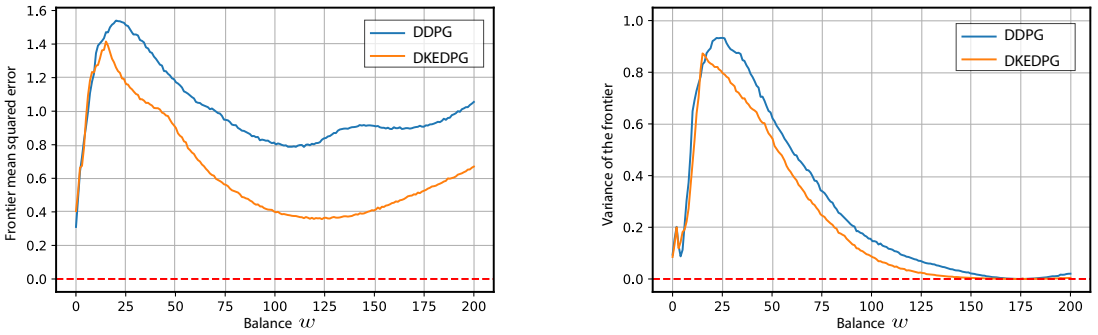
	DPG	DKEDPG
Interpretability index	89.37%	99.46%
Systemic consistency index ($C_{95\%}/C_{99\%}$)	22%/ 0%	100%/ 100%
Averaged MSE of the learned frontier	1.093	0.648
Averaged variance of the learned frontier	0.284	0.228

Table 3.1: Average performance summary at learning termination

frontiers $\hat{\lambda}^{\text{DDPG}}(w)$ and $\hat{\lambda}^{\text{DKEDPG}}(w)$ using our 50 independent runs. Additionally, in Fig. 3.5b we use our knowledge of theoretically optimal frontier $\lambda^\star(w)$ to compute the variances of $\hat{\lambda}^{\text{DDPG}}(w)$ and $\hat{\lambda}^{\text{DKEDPG}}(w)$, respectively. From Fig. 3.5a, we observe a noticeable reduction in MSE (on average 0.4, see table 1) when balance w is not too small. From bias-variance decomposition of MSE, part of this reduction is due to reduction in the variance and the rest is due to reduction in the bias. Fig. 3.5b indicates that most of the reduction in MSE is due to reduction in the bias. This is because the average reduction in the variance is roughly 0.05 (see table 1) which only captures 12% of the reduction in MSE.

3.5 Conclusion

Domain-knowledge enhanced reinforcement learning naturally incorporates structural insights and principles, thus enabling the learning of interpretable policies. The domain expertise is thereby formulated in terms of monotonicity constraints on the policy, and is incorporated into the learning algorithm using a barrier regularizer that imposes penalties for policy violations. Our results demonstrate that penalizing for the monotonicity does not impact learning speed, convergence or performance; on the upside, it provides quantifiable guarantees of interpretability in the policy space.



a) b)

Figure 3.5: a) MSE of the action frontier (when fixing w) computed over 50 independent runs $\hat{\lambda}^{\text{DDPG}}(w)$ and $\hat{\lambda}^{\text{DKEDPG}}(w)$. b) Variance of the action frontier (when fixing w) computed over 50 independent runs using the theoretically optimal action frontier (see Fig. 3.4).

Societal Implications and Broader Impact

In contrast to a theoretical reinforcement learning setting where an agent interacts directly with the learning environment to produce policy updates using quick simulated feedback, in many practical applications a learned policy is subject to a human decision maker's oversight and will need to be validated in a real-world setting. Thus, outside a lab environment, a decision maker needs consistency—even at the price of somewhat suboptimal performance, for this provides not only interpretability and understandability as mentioned at the outset, but also forms the basis of *auditability*. That is, provided complete interpretability (and systemic consistency) of a learned policy, the decision maker is able to explain the policy to a third party (including a benevolent court of law if necessary) and can therefore provide a clear rationale (be it *ex ante* or *ex post*) for the implementation of machine-learned actions. In the setting of a stochastic control problem with asynchronous rewards we have shown that interpretability regularization, that is the inclusion of penalty terms for deviations from policy shape constraints, may guide the learning agent to fully interpretable policies. To quantify the generic suitability of a learned policy, we have proposed two measures, namely an interpretability index (as percentage of shape-constraint adherence on the learned action set) and a systemic consistency index, which measures interpretability at a defined point of policy convergence. The hope is that these results may contribute to the reduction of the "lawlessness of machine algorithms" by allowing external parties to verify objective measures of interpretability and systemic consistency. In this way, the paper contributes to the broader discussion on ethical machine learning and its implications for business applications.

Application Part III

4 Quantifying Endogeneity of Cryptocurrency Markets

This chapter showcases one of the salient applications of Hawkes processes in high-frequency finance; it is based on Mark, M., Sila, J., and Weber, T. A. (2020b). Quantifying Endogeneity of Cryptocurrency Markets. *European Journal of Finance*. DOI: 10.1080/1351847X.2020.1791925.

4.1 Introduction

Bitcoin, introduced by Nakamoto (2008), is arguably one of the most interesting financial innovations of this century. Without any central authority an *ad hoc* peer-to-peer network issues a tradeable asset that can be considered an alternative to fiat currencies, with all the necessary features such as value storage and fungibility (as a medium of exchange). Moreover, cryptocurrencies offer certain advantages, such as fast and low-cost execution—particularly when compared to traditional financial institutions. The underlying blockchain technology renders the recorded transactions public and transparent. Meanwhile, the cryptocurrency space has spawned thousands of Bitcoin-like digital assets, creating a financial platform akin to foreign-exchange markets for fiat currencies. Yet, with a market capitalization of about $200 billion its size is still fairly insignificant in comparison with the $20 trillion invested in the S&P 500 stock index. Current public discussion tends to focus on certain technical or legal points. Regarding the place of Bitcoin in the current financial system, Baur et al. (2018) conclude that Bitcoin is a speculative asset and thus far has not served as an alternative currency or medium of exchange. Kristoufek (2015)—using a wavelet analysis—recognizes Bitcoin as a hybrid asset whose price is influenced by money supply and adoption in trade, quite in accordance with standard economic theory. As of now, our understanding of cryptocurrencies and their place within the traditional monetary system remains sketchy. Despite the fact that crypto-market capitalization has grown significantly, the market itself has been almost entirely unregulated. The lack of a centralized regulatory body, together with extreme market swings, has given rise to much criticism and caution. On the other hand, the apparently functioning crypto-market presents a natural experiment introducing one of the most laissez-faire financial exchanges of all time, which invites research on market dynamics and investment behavior. Indeed, cryptocurrency exchanges offer unprecedented public access to market data, thus allowing

for in-depth analyses and comparisons to the theory describing their traditional counterparts.

Conceptually, one can think of financial markets as devices to convert information and beliefs about underlying fundamentals into prices (Grossman, 1989). A critical question, that still remains largely unanswered (not only for cryptocurrencies), is how much prices are indeed driven by observable information. In other words, are markets sufficiently immediate and efficient to track rapid changes of security valuations with commensurate price adjustments? According to the strong version of the Efficient Market Hypothesis (EMH), prices are a perfect reflection of available news (Fama, 1970). That is, markets are driven exogenously: [1] any new information is instantaneously absorbed and reflected in a new equilibrium price. If this is the case (and investors are rational), a market crash can arise only as a consequence of a negative high-impact news release or when significant indicator thresholds are crossed (Reinhard and Rogoff, 2009). However, global financial markets have witnessed multiple flash-crash events in which vast amounts of capital were lost and again recovered in a matter of minutes, without any clear exogenous trigger. Using high-frequency data, Bouchaud (2009) concludes that merely a small fraction of significant price jumps can be explained by exogenous events. Furthermore, empirical anomalies do not conform with the neoclassical framework, such as the "excess volatility puzzle" where prices move more than would be justified by the pertinent news flows (LeRoy and Porter, 1981; Shiller, 1981). Even in the cryptocurrency research community the EMH has been a source of persistent controversy. For instance, Kristoufek (2018) and Urquhart (2016) conclude that bitcoin markets are close-to-efficient. By contrast, Jiang et al. (2018), as well as Vidal-Tomás et al. (2019), obtain empirical results which tend to contradict the EMH. The latter evidence suggests the presence of a significant endogenous component, especially on smaller timescales, critical for the price evolution of digital currencies. Thus, one expects that overall price dynamics must be driven by a time-dependent complex interplay of exogenous *and* endogenous factors. Collective behavioral phenomena, such as herding or imitation (Hong et al., 2005; Lux, 1995), offer some plausible explanations. Indeed, Bouri et al. (2019) and Ajaz and Kumar (2018) provide empirical evidence for herding behavior in cryptocurrency markets. However, their respective methodologies—based on cross-sectional standard deviations of daily returns—are impervious to the more granular intra-day trading events. Additional sources of the unexplained endogeneity are frequently attributed to strategic order splitting, margin calls, stop-loss triggers, or high-frequency traders.

Recently, a class of self-exciting point processes was recognized as a suitable tool for disentangling and quantifying the underlying dynamics of the price process, as one of its inherent features is a neat separation of the endogenous and exogenous action triggers. The discussion on endogeneity and its evolution in the markets (i.e., the endo-exo problem) was broached by Filimonov and Sornette (2012), who fit a univariate exponential Hawkes process to E-mini S&P 500 futures traded between 1998 and 2010. They discuss reflexivity[2] on a micro scale (i.e., in

[1] At this point, we disregard systemic general-equilibrium feedback effects based on the fact that price levels impact investors' budgets, which in turn changes their consumption behavior, thus generically influencing the prices of the underlying assets (as in the well-known "Ford effect"), albeit on a slow timescale.

[2] Market reflexivity is a term coined by Soros (1994) highlighting the positive feedback mechanism where

intervals of less than 1 hour) and report a significant increase in the level of endogeneity over the observed period (from 0.3 in 1998 to 0.8 in 2012), which can be attributed to the rise in algorithmic trading. Hardiman et al. (2013) ("HBB") revisit this problem with the same dataset; however, instead of the fast decaying exponential, they opt for a heavy-tail long-memory power-law kernel. While agreeing about the rise in the short-term reflexivity, the authors conclude that markets were in fact persistently operating around the criticality level. Their corollary is that price dynamics are therefore best described by two separate kernels, one for long- and one for short-term memory—thus taking into account the meso market structure (in intervals of about 1 day). This argument has been refuted by Filimonov and Sornette (2015) who identify numerous estimation-related issues in HBB's methodology, such as an upward bias in the presence of outliers for the power-law kernel norm, whence questioning HBB's results. To settle the discussion Hardiman and Bouchaud (2014) develop an empirical estimator of the branching ratio, further supporting their claims of market criticality and the presence of long-memory properties. However, such a moment-based estimator attributes all the dispersion beyond the homogenous Poisson process to the branching ratio. Consequently, when fitted to empirical financial data, infamous for trends, data artifacts, and other non-stationarities, the branching ratio is expected to be noticeably biased upward. In fact, Wehrli et al. (2021) showed that any data set that exhibits long-range dependence according to second-order properties yields a critical branching ratio estimate by construction. Wheatley et al. (2019) addresses the problem of data non-stationarities by utilizing a flexible base rate function based on B-Splines. Their results suggest that although the market mid-price changes are strongly self-excited, the criticality is strongly rejected, at least for the univariate microstructure Hawkes model. The methodology framed in this paper closely follows the one of Filimonov and Sornette (2012) and therefore is aimed at measuring and investigating endogeneity on the micro-scale level (i.e., $\lessapprox 1$ hour).

Despite cryptocurrencies being an active field of research, to the best of our knowledge, as of yet there has been no discussion on the origins of price dynamics. In this paper, we quantify the degree of market endogeneity and investigate its temporal dependence. This allows us to estimate the market susceptibility (and inefficiency due) to endogenous behavioral biases (such as herding) as well as its reactiveness to exogenous shocks. We also uncover structural similarities in the price dynamics of cryptocurrencies, equities, commodities, and foreign exchange (FX). The findings shed light on the nature of Bitcoin from the vantage point of self-exciting point processes.

Our paper proceeds as follows. Section 4.2 motivates and introduces the Hawkes model along with its branching-structure representation which allows us to characterize the endo-exo dynamics. Section 4.3 follows with a description of the dataset. Section 4.4 presents the key findings, and Sec. 4.5 concludes.

investors' anticipation leads to self-fulfilling prophecies, just as in a Keynesian beauty contest (Keynes, 1936).

4.2 Model

4.2.1 Motivation

Before diving into the methodology of self-exciting point processes, it is useful to pinpoint the salient deficiencies of standard models which do not capture the endogenous component in the price dynamics. Consider a finite sequence of trade times $(T_1, T_2, \ldots, T_n)$. Disregarding the direction and volume of the trades, and assuming independence among them (i.e., every trader acts independently based on private information), this description corresponds to a perfectly exogenous market without behavioral biases such as imitation or herding. Therefore, for sufficiently small time intervals (to account for intra-day structural breaks) the observed order arrivals should reasonably well follow a homogenous Poisson point process (HPP). Fig. 4.1 shows an empirical realization of trade arrivals (in a 30-minute window), next to a simulated sequence of Poisson arrivals at the same rate. It can be easily seen that an HPP does not capture the essence of the empirical sequence, since observed market orders exhibit marked clustering—quite in contrast to a memoryless Poisson process.[3]

A natural extension of an HPP is a Markov-modulated Poisson process (MMPP), as an instance of a "hidden Markov model." An MMPP is a doubly stochastic point process whose intensity depends on the (unobserved) state of a Markov process. MMPPs have been applied successfully to return data in view of identifying structural breaks and distinguishing different volatility regimes (Engel, 1994). However, when setting up an MMPP, a critical decision has to be made at the model-specification stage on the *number* of hidden states, for example, with help of information criteria. In our dataset, a reliable calibration of an MMPP proved infeasible: our fits indicated an improvement with an increasing number of hidden states (to more than 10) with respect to both AIC *and* BIC,[4] suggesting a fragmented process with no clear state separation.

Remark 5 Using the CSR (complete spatial randomness) framework described by Clark and Evans (1954) one can strongly reject the null hypothesis of spatial randomness, practically for any time horizon (ranging from mere minutes to full days). This finding is in line with the barcode in Fig. 4.1a, which features heavy clustering.

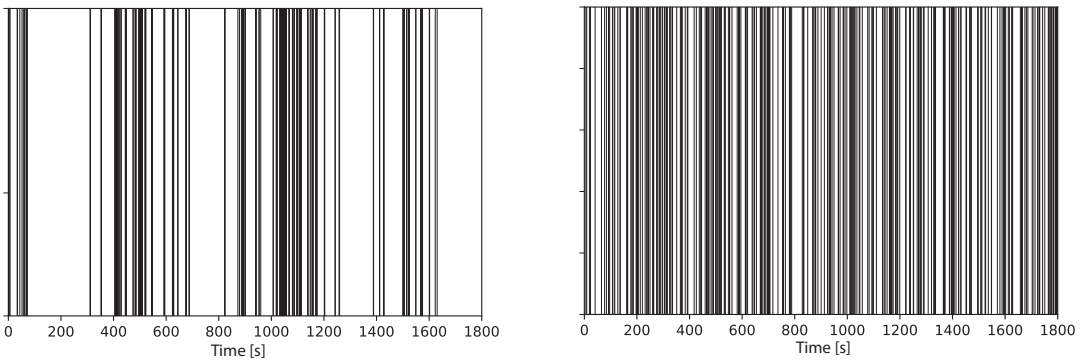
4.2.2 Univariate Hawkes Process

A univariate Hawkes process is a linear self-exciting point process with a conditional intensity function, defined as

$$\lambda(t|\mathscr{H}_t) = \mu(t) + \int_0^t g(t-\tau)\, dN_\tau = \mu(t) + \sum_{i:\tau_i<t} g(t-\tau_i), \tag{4.1}$$

[3] Using our dataset, we fitted HPPs to 30-minute intervals of a given trading day, and find that HPPs are ill-suited for the description of order arrivals. In particular, all residual tests in Sec. 4.2.4 failed.

[4] AIC: Akaike Information Criterion; BIC: Bayesian Information Criterion.



a) b)

Figure 4.1: Barcode plots. a) Empirical mid-price change times over 30 minutes (267 arrivals). b) Simulation of a homogenous Poisson point process with intensity estimated from the empirical data (290 arrivals).

where $\tau_i \geq 0$ denotes the i-th arrival time (for $i \geq 1$). The baseline intensity $\mu(t)$ is a deterministic function of time, while $g : \mathbb{R}_+ \to \mathbb{R}_+$ is a (nonnegative-valued) self-excitation function (or "memory kernel") which determines the covariance properties of the process. The filtration $\mathscr{H}_t$ describes the estimation-relevant record of the process history and for our purposes includes all event-arrival times on the interval $[0, t)$.

In the framework of Hawkes processes, "endogeneity" refers to the ability of generating new arrivals from past events. This notion of endogeneity can be reinforced and formalized using an alternative but equivalent view of the process in terms of a (stochastic) branching structure (Hawkes and Oakes, 1974). The latter provides for a direct mapping between arrivals and clusters, where each cluster starts with an immigrant generated from an inhomogeneous Poisson process with baseline intensity μ. Consistent with Eq. (4.1), every arrival triggers a spike in the intensity through the memory kernel, thus generating its own offspring arrivals according to an inhomogeneous Poisson process of intensity g. This cascades through all offsprings, effectively creating a hierarchical branching structure. Eventually, every event can be labeled either as an "immigrant," generated exogenously through the deterministic drift without an existing parent, or as an endogenously created "offspring." The central parameter controlling the size of endogenously generated offspring families, generally referred to as the "branching ratio" n, is defined as the average number of offsprings per event:

$$n = \int_0^\infty g(t)\, dt. \tag{4.2}$$

Conceptually, the branching ratio determines the degree of self-excitation in the process. The latter captures the percentage of the arrivals that are generated endogenously—as a consequence of previous arrivals. Based on the value of the branching ratio, one can distinguish four different regimes:

(i) $n = 0$ (memoryless): inhomogeneous Poisson point process which features exclusively immigrant arrivals;

(ii) $n < 1$ (sub-critical): nonexplosive process;[5]

(iii) $n = 1$ (critical): marginally unstable (or stable) process where a generation of offsprings lives indefinitely (the existence of such processes was proved by Brémaud and Massoulié (2001) for $\mu(t) = 0$);

(iv) $n > 1$ (super-critical): nonstationary explosive process with finite intensity but infinite/nonintegrable covariance. (In other words, a single event starts an infinite family, and the process explodes.)

This fourfold separation provides a succinct partition for the endo-exo market dynamics, similar in interpretation to such distinctions in the context of well-established autoregressive processes. Ultimately, it allows us to describe the endogeneity by a characteristic number, comparable across asset classes and financial instruments.

When inferring the branching ratio from data one has two main options: *either* stochastic declustering (Zhuang et al., 2002) (which essentially requires reverse-engineering of the clusters) *or* maximum-likelihood estimation (MLE) in combination with Eq. (4.2). In the remainder of the paper, we pursue the second approach for its relative simplicity and clarity.

4.2.3 Parametric Kernels

The two most prominent classes of (parametrized) self-excitation functions are "exponential" and "power-law."

(a) Exponential kernel:

$$g(t) = \frac{n}{\beta} \exp\left[-\frac{t}{\beta}\right], \quad t \geq 0, \tag{4.3}$$

where $n \geq 0$ denotes the branching ratio and $\beta > 0$ is the decay parameter. The exponential kernel satisfies the Markov property, rendering it particularly popular. Furthermore, as shown by Ogata (1981), it allows for recursive maximum-likelihood computations which reduce the computational complexity from $\mathcal{O}(N_T^2)$ to $\mathcal{O}(N_T)$.

(b) Power-law kernel:

$$g(t) = \frac{n\varepsilon\tau_0^{\varepsilon}}{(t+\tau_0)^{1+\varepsilon}}, \quad t \geq 0, \tag{4.4}$$

where $n \geq 0$ denotes the branching ratio; the shift parameter $\tau_0 > 0$ ensures the integrability of the kernel, and $\varepsilon > 0$ sets the decay speed. This particular form gained its

[5] This holds, as long as the immigrant generation process $\mu(t)$ remains bounded.

popularity through the *epidemic-type aftershock sequence* model of earthquake occurrences.[6] Compared to the exponential kernel, the power-law kernel features a "long memory" (due to its relatively thick tail) that may be better suited for financial markets than the exponential kernel. Additionally, there exist other variations of the power-law kernel that differ in the way the regularization is introduced. The primary reason for using this class of kernels is the fat tail shared by its various members (at least approximately; see Remark 6).

Remark 6 Applying directly the power-law kernel as specified in Eq. (4.4) carries a significant computational burden, as Ogata's recursive MLE formulation (Ogata, 1978) is not available. In order to expedite computations, we consider an approximation constructed from a power-law-weighted sum of exponential kernels (Bochud and Challet, 2007)

$$g(t) \approx \hat{g}(t) \equiv \frac{n}{Z} \sum_{k=0}^{M-1} a_k^{-(1+\varepsilon)} \exp\left[-\frac{t}{a_k}\right], \quad t \geq 0, \tag{4.4'}$$

where $a_k = \tau_0 m^k$. The power-law substitute $\hat{g}$ is parametrized by the branching ratio n, the tail exponent ε, and the shift parameter τ_0. The additional tuning parameters $m > 1$ (best chosen not too large to avoid an excessive saw-tooth pattern for small t) and $M > 1$ impact the quality and range of the approximation. This approximation, employed by HBB, captures long-term dependencies while leveraging a recursive formulation of the maximum-likelihood function, resulting in reduced fitting times. In fact, the formulation allows to accurately reproduce a power-law decaying kernel $g(t) \sim t^{-1-\epsilon}$ for $\tau_0 = a_0 \ll t \ll a_{M-1}$ (Bouchaud et al., 2018). The parameter Z is chosen such that n is equal to the true branching ratio of the kernel, $\int_0^\infty \hat{g}(t)\, dt = n$, i.e., $Z = \sum_{k=0}^{M-1} (\tau_0 m^k)^{-\varepsilon} = \tau_0^{-\varepsilon}(1 - m^{-\varepsilon M})/(1 - m^{-\varepsilon})$.

In what follows, all references to a power-law kernel refer to the substitute $\hat{g}$ in Eq. (4.4'), with tuning-parameter values $(m, M) = (2, 10)$. For more discussion about kernel properties and their differences, see the comprehensive review by Bacry et al. (2015).

4.2.4 Goodness-of-Fit Tests

A standard method for assessing the quality of a point-process fit is the "residual analysis," which consists of computing the time-deformed series of durations $\{\xi_i\}_{i=1}^\infty$ using the estimated conditional intensity $\hat{\lambda}(s|\mathscr{H}_T)$,[7]

$$\xi_i = \int_{\tau_{i-1}}^{\tau_i} \hat{\lambda}(s|\mathscr{H}_T)\, ds,$$

[6] The power-law kernel in Eq. (4.4) corresponds in fact to a (renormalized) Pareto-distribution.

[7] The estimated conditional intensity $\hat{\lambda}(\cdot|\mathscr{H}_T)$ is obtained via simulation of Eq. (4.1) using maximum-likelihood parameters, conditional on the available data $\mathscr{H}_T$ on the observation interval $[0, T]$.

and then statistically testing it for theoretical properties. In the case where a Hawkes process presents a fairly accurate description of the empirical data, residuals of the interarrival times are independently and identically distributed (i.i.d.) draws from an exponential distribution with parameter $\lambda = 1$. We assess three theoretical properties using the standard statistical tests, as follows:

(A) Ljung and Box (1978) test (LB) for the *absence of autocorrelations* to ensure independence of residuals, using up to 20 lags;

(B) Kolgomorov-Smirnov test (KS) for the *distance between the empirical and the theoretical distribution* of the residual process;

(C) Engle and Russell (1998) test (ED) for *excess dispersion* in the residuals.

In our setting, a parametrized model that passes all three tests simultaneously is considered "successful" and considered a viable explanation of the observed data.

Remark 7 (Brock-Dechert-Scheinkman (BDS) Independence Test) The triade of statistical tests (A), (B) and (C) constitutes quite a standard testing suite for Hawkes-process residuals. For the sake of robustness, we performed an additional BDS test of independence (Brock et al., 1996). Given that this additional test did not change the test-survival statistics by much, we argue that a check of the independence hypothesis can be captured satisfactorily by the LB autocorrelation test.

4.3 Data

Our data set includes all executed transactions on the BitMEX cryptocurrency exchange between March 1 and May 1 of 2019. BitMEX was selected as the largest crypto-exchange in terms of its trading volume, particularly with respect to Bitcoin (BTC) contracts settled in USD. The trading is open 24 hours a day, so that it closely resembles traditional FX markets. Each trade is recorded with its corresponding time stamp, volume, price, and whether or not the transaction changed the last transaction price (resulting in an uptick or downtick). The available millisecond resolution in the data presents the highest available granularity for this market.
Even though our dataset tracks all BitMEX-traded instruments, including exotics such as Cardano (ADA) or Tron (TRX), we restrict attention to Bitcoin contracts (ticker XBTUSD), since it accounts for the vast majority of trading volume; see Fig. 4.2.[8]

[8] A full description of all available contracts can be found at www.bitmex.com. BitMEX was founded in 2014; it is currently owned and operated by HDR Global Trading Ltd.

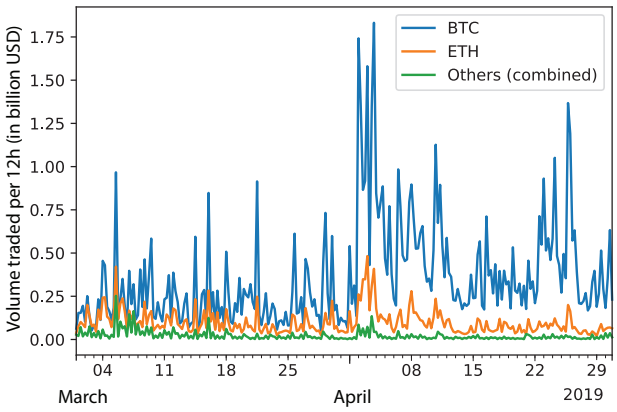


Figure 4.2: On any given day, the number of transactions (or Hawkes-process arrivals) differs greatly among currencies. Bitcoin contracts account for almost two thirds of the trading activity, and together with the Ethereum market, it accounts for practically all trades. The average number of arrivals on the XBTUSD market is almost an order of magnitude larger than for other currencies, with a peak daily activity of around 1 million recorded trades.

timestamp	ordertype	volume	price	ticktype	arrival
2019-02-02 22:45:58.560	Buy	20	3433.5	PlusTick	0.000
2019-02-02 22:46:03.493	Sell	10	3433.0	MinusTick	4.933
2019-02-02 22:46:06.754	Sell	50	3433.0	ZeroMinusTick	8.194
2019-02-02 22:46:09.639	Sell	4	3433.0	ZeroMinusTick	11.079
2019-02-02 22:46:10.679	Buy	21	3433.5	PlusTick	12.119

Table 4.1: Description of the market-order data from the BitMEX exchange, which accounts for about 10 percent of the entire BTC trading volume.

4.3.1 Measures of Market Activity

The precise definition of the "events" to be considered is not only critical for the ex-post confidence in the identification of the arrival process, but also for the informational value the estimated coefficients may carry. It is therefore imperative to select a reliable measure of market activity which is robust to the "microstructure noise" omnipresent in high-frequency data. In the finance literature thus far, Hawkes processes have been fitted mainly to the most granular (and noisy) trade data or to various "price actions" (i.e., movements) near the "best prices" (i.e., $a(t)$ or $b(t)$; see below). Although the trade-arrival rate may, at first glance, seem to be a reasonable metric for market activity, it does come with an important drawback, as not all trades are equal in their impact due to their quite disparate volumes. To take trade sizes into account, one would need to consider a "marked" version of a Hawkes process which is significantly more intricate to fit.[9] Practitioners commonly track four different quotations (as

[9]While the inclusion of i.i.d. marks, independent of the event-arrival distribution, is a straightforward extension of a standard Hawkes-process estimation (Chehrazi and Weber, 2015), identification without such an

a function of time t), each giving rise to a price action and serving a different purpose:

- best-bid $b(t)$;
- best-ask $a(t)$;
- last transaction price $p_{\text{tr}}(t)$;
- mid-price $p_m(t) = (a(t) + b(t))/2$.

Best-bid and best-ask reflect the upper and lower price boundaries of the standing limit orders (Fig. 4.3a), at which a trader can immediately engage in selling or buying (with a market order), respectively, up to the cumulative volume of standing orders in the limit-order book (LOB) at the given price level. They can be regarded as proxies for the market makers' supply and demand.

4.3.2 Mid-Price Tracking

When a buy (resp., sell) market order arrives on an exchange at time t, it is paired with the best-ask (resp., best-bid) price available, completing a trade that produces a last transaction price $p_{\text{tr}}(t)$. As trades arrive in random order, with the direction of the trade being a random variable as well, the last transaction price jumps sporadically—at times even without concomitant change in supply or demand. This behavior is referred to as "bid-ask bounce," and it has been established as a proper noise source in its own right (Aït-Sahalia and Yu, 2008; Black, 1986). Hence, the mid-price is regarded as a more reliable proxy for asset values than the aforementioned best prices (i.e., $a(t)$ and $b(t)$), particularly because it does not suffer from the bid-ask bounce; Fig. 4.3b shows that $p_m(t)$ is much less noisy than $p_{\text{tr}}(t)$. A change in mid-price can arise due to one of the following three reasons:

(I) Cancellation of an existing limit order at the best-bid/ask price;

(II) Submission of a new limit order at a new best-bid/ask price;

(III) Depletion of the available LOB volume at the best-bid/ask price by market orders.

Even though causes (I) and (II) result from limit orders submitted by liquidity providers (who want to trade), the publicly visible LOB does not reflect the true supply and demand in the market. This comes as a consequence of market participants' (particularly large liquidity providers') reluctance to disclose private information by openly displaying their intentions and intended future positions. Consequently, in fast markets, such as the Bitcoin exchange considered here, a large portion of cancelled orders and new limit orders represent so-called *ghost* orders (Lewis and Baker, 2014) whose main purpose is to pry for private information.

i.i.d. assumption—as needed for trade volumes in relation to the trade-arrival process—is still an open problem.

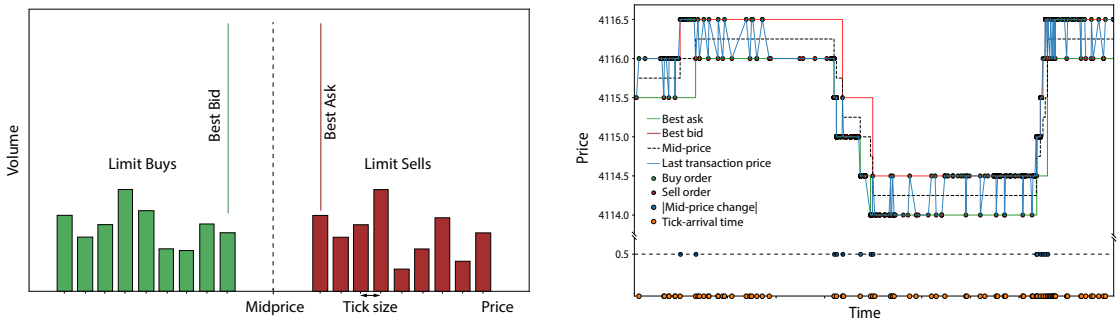


Figure 4.3: Illustrative model of the limit-order book (LOB) a) and construction of the mid-price from transactional data (b).

Indeed, in our dataset cancellations and limit orders account for about 15% of all mid-price changes. We argue that—unlike the first two—it is cause (III) that principally reflects the actual interaction of supply and demand, and as such it should be considered the most reliable information source.

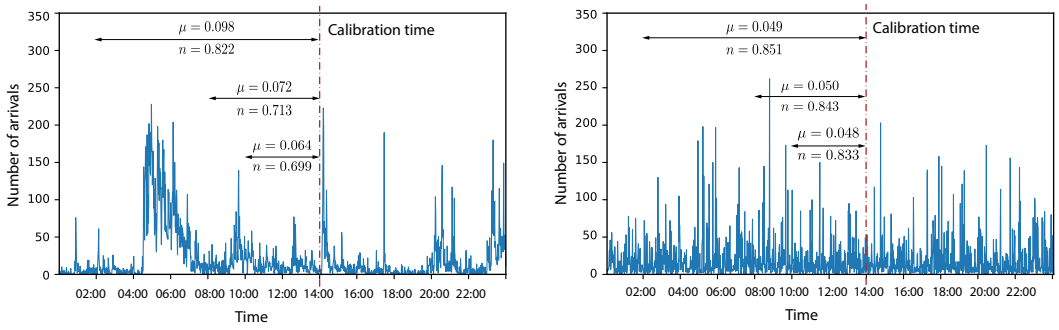
4.4 Results

Tracking Bitcoin mid-price changes (caused by filled orders between March 1 and May 1, 2019), we now fit a univariate Hawkes process, defined in Eq. (4.1), with exponential and power-law kernels, as in Eqs. (4.3) and (4.4), using maximum-likelihood estimation. MLE is our technique of choice for the identification of Hawkes processes, which amounts to solving the following optimization program:

$$\begin{aligned} \max_{\theta \in \Theta} \quad & \log \mathscr{L}\left(\theta | \mathscr{H}_T\right), \\ \text{subject to} \quad & \theta \geq 0, \end{aligned} \tag{M}$$

where θ represents a vector of kernel and base-rate parameters. The likelihood function, derived by Rubin (1972), is asymptotically normal, efficient, and consistent (Ogata, 1978). As such, it constitutes a straightforward statistical inference technique for the family of self-exciting point processes. The flipside is that its nonconvexity in the decay parameter (β for the exponential and ε for the power-law kernel, respectively), coupled with an extreme flatness of the log-likelihood surface near the optimum (Veen and Schoenberg, 2008), makes reliable calibration a challenging task (Mark and Weber, 2020). In order to circumvent these problems we solve the optimization program (M) in parallel, for a batch of 500 starting guesses and then single out the estimation result for the parameter vector which attains the highest log-likelihood.

Next, we consider the—from a practitioner's standpoint—important question (neglected in the literature thus far) of how to determine the optimal observation length T for an effective estimation of the point-process model in Eq. (4.1). After this, we are ready to construct a



a) b)

Figure 4.4: Impact of a differing estimation horizon $T \in \{4, 6, 12\}$ hours on the estimate's accuracy. In contrast to simulated data coming from a single point process realization, empirical data features intra-day seasonalities that significantly affect the maximum-likelihood estimates. a) Empirical data from April 1, 2019. Total number of arrivals: $N_T = 30{,}683$ (for $T = 12$). b)Data simulated from an exponential Hawkes process with parameters $\mu = 0.05$, $\beta = 1$ and $n = 0.85$. Total number of observations: $N_T = 27{,}540$ (for $T = 12$).)

reflexivity index for cryptocurrency markets (so as to quantify their endogeneity).

4.4.1 Optimal Estimation Horizon

When deliberating about a most preferred observation horizon T, at least for a stationary process with constant parameters a greedy approach (i.e., pursuing "more is better") would indeed appear to be successful. However, in a real-world situation, the empirical trade data at hand most likely would not derive from a single, long, and historically consistent generating process. Thus, a more considerate and appropriate view would be to account for switching regimes, thus allowing conceptually for a data history generated by a concatenation of materially different processes. This problem of detecting the corresponding phase-transition times, commonly known as the Poisson disorder problem (Peskir and Shiryaev, 2002), has been studied in the context of homogeneous Poisson processes and unfortunately does not have a straightforward extension to self-exciting processes. Therefore, one has to carefully calibrate the length of the estimation windows such that the history contains a sufficiently large sample for obtaining accurate estimates, while at the same time avoiding a calibration across multiple regimes. On one hand, an inference from a shorter window plays into the assumption of a constant base rate, narrowing the view sufficiently to be able to negate the empirical regularity that mid-price changes feature marked intra-day seasonalities. On the other hand, short estimation windows do limit the kernel's memory and thus disregard interdependencies across time, which develop over hours, days, or even longer periods of time. This interplay of phenomena is illustrated in Fig. 4.4 which compares fits over various observation horizons, for both simulated and empirical data.

In order to resolve this tradeoff, we rely on a robust numerical experiment. As discussed

above, we try to identify a minimal window size T such that the number of observations is sufficient for an "accurate" inference. Consider a family of exponential Hawkes processes with a moderate branching ratio $n = 0.5$ but variable baseline intensity $\mu \in [0, 0.2]$ representing different market regimes.[10] We perform an estimation of each process, given observation horizons $T \in [60, 10800]$,[11] and we measure the relative estimation error $e = \frac{\|\hat{\theta}-\theta\|_2}{\|\theta\|_2}$, where $\|\cdot\|_2$ denotes the standard Euclidean distance (2-norm). To ensure robustness of the experiment, we obtain a mean relative error for every individual horizon T (using Monte Carlo simulation over 1,000 process realizations). Fig. 4.5a depicts the relationship between the observation horizon T and relative error e, obtained as a mean across all simulation paths, together with bootstrapped (5%/95%)-confidence intervals (Efron and Tibshirani, 1994).

As expected, higher μ-regimes can handle shorter observation horizons—without a significant impact on accuracy. The most preferred (i.e., "optimal") horizon, denoted by T^α, is selected such that for all $T \geq T^\alpha$, $e_{0.95}(T) < \alpha$, where α is a given acceptance threshold.[12] In other words, *the optimal observation horizon is the minimal horizon beyond which the relative error does not exceed α at a 95%-confidence level.* Fig. 4.5b illustrates the determination of this threshold for various values of μ, resulting in a effective decision tool for determining the optimal observation horizon conditional for a given baseline intensity. More specifically, consider the realization of a self-exciting point process on $[0, T]$, formed as a concatenation of $K \geq 1$ Hawkes processes with different regime parameters (μ_k, n_k), for $k \in \{1, \ldots, K\}$, each lasting T_k, so $\sum_{k=1}^K T_k = T$. The goal is to estimate this process on a rolling horizon, using an observation window of smallest possible length T, so as to prevent averaging the fit over multiple regimes. We approximate the mean intensity of the compound process as follows:

$$\Lambda = \sum_{k=1}^{K} \frac{T_k}{T} \Lambda_k \approx \frac{\#\text{events on } [0, T]}{T}; \tag{4.5}$$

using the identity for the average intensity (Hawkes, 1971b), together with a reasonable upper bound $\bar{n}$ for the branching ratio n. We thus recover

$$\underline{\mu} = \Lambda(1 - \bar{n}), \tag{4.6}$$

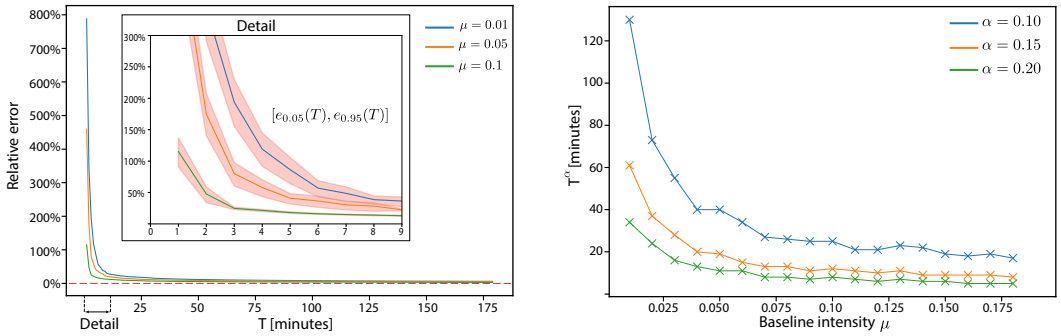
which constitutes an approximate lower bound for the baseline mean μ. Together with the downward-sloping dependence of the relative error e on T (depicted in Fig. 4.5b for $\alpha \in \{10\%, 15\%, 20\%\}$), the latter yields a minimal observation horizon, which we refer to as "optimal estimation horizon" T^α, *as it most effectively trades off the data requirements for estimation against the volatile nature of regime changes in the trading activity.*

Remark 8 Although the preceding analysis is purely simulation-based, the results pinpoint

[10] Indeed, market shifts tend to manifest themselves in μ rather than in the mode of self-excitation (so g remains fixed), as suggested by our data and Wheatley et al. (2019).

[11] This corresponds to a range of time horizons between 1 minute and 3 hours.

[12] The observed relative errors $e_{0.05}(T)$ and $e_{0.95}(T)$, over the observation interval $[0, T]$, mark the deviations relative to 5% and 95% of the data, respectively; see Fig. 4.5a.



a) b)

Figure 4.5: Monte Carlo analysis of the relative error for various baseline regimes. Although Fig. 4.4 was constructed using a fixed self-excitation parameter $n = 0.5$, the results serve as a conservative decision tool for the optimal horizon T^{α}, as higher n-values translate into more observations and thus faster and more reliable calibration. The value was not chosen arbitrarily; it is the lowest self-excitation measured on nonoverlapping 10-minute windows using the approximate branching-ratio estimator (Hardiman and Bouchaud, 2014). a) Relative error as a function of the time horizon, for representative base-rate regimes. Red-shaded area represents bootstrapped mean confidence intervals. b) Relationship between optimal (minimal) observation horizons and different μ-regimes for three representative acceptance thresholds.

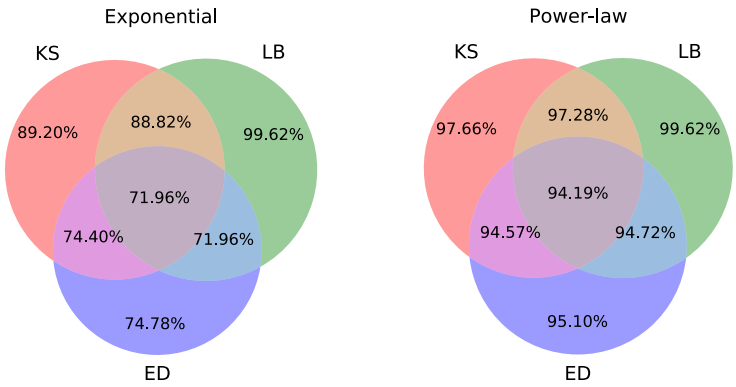
high-quality approximations. Indeed, to the best of our knowledge the given simulation-based method provides the first effective work-around for the (as of yet unsolved) "Hawkes disorder problem." The practical significance of this problem can be gauged by inspecting Fig. 4.9, which shows a typical section of historical price movements and volatility—featuring two distinct regimes of activity. We also note that the empirical results were derived using a Hawkes process with exponential kernel. They readily extend to the power-law formulation in Eq. (4.4'), provided that the tail of the kernel does not significantly exceed the observation window. g

4.4.2 Reflexivity Index

Building on the analysis from the previous subsection, we now determine the adequate look-back period first by measuring the mean intensity, $\Lambda = \frac{416{,}019 \text{ events}}{5{,}184{,}000 \text{ s}} = 0.08$. Next, we recover the baseline intensity, $\underline{\mu} = \Lambda(1 - \bar{n}) = 0.16$, which corresponds approximately to a 60-minute look-back period T for a 15% relative error; see Fig. 4.5b.[13] For direct comparison with the study conducted by Filimonov and Sornette (2012) on S&P 500 E-mini contracts, we calibrate the process for both parametric kernels described in Sec. 4.2.3 with the additional look-back periods of 10 and 30 minutes, on a minute-by-minute rolling horizon.

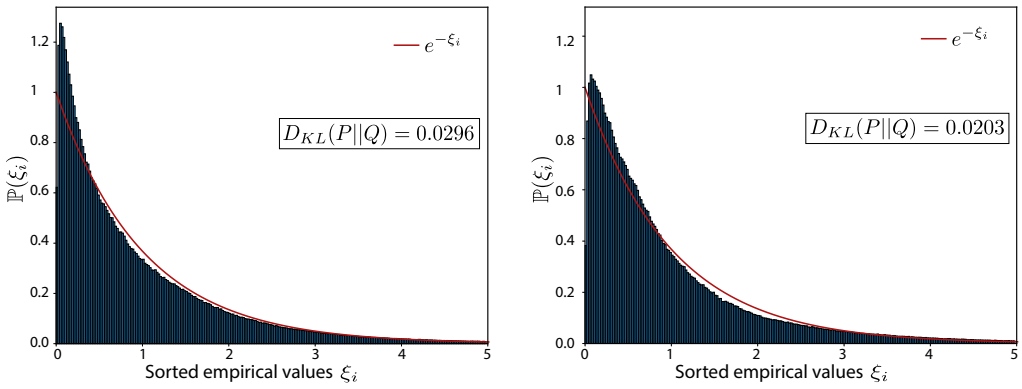
To judge the significance of the results, we refer to Table 4.2, which contains pass rates for the

[13] Value $\bar{n} = 0.8$ was chosen, for it is the mean value of the branching ratio measured on nonoverlapping 10-minute windows using the approximate branching-ratio estimator (Hardiman and Bouchaud, 2014).



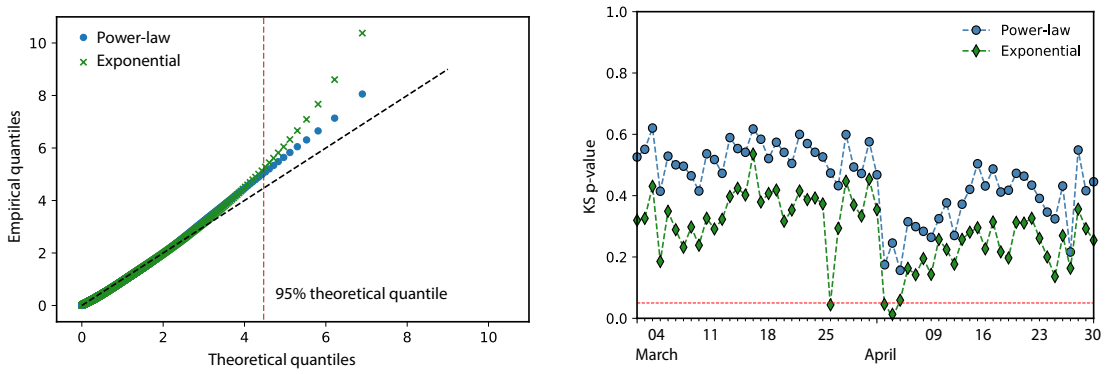
a) b)

Figure 4.6: Fractions of significant fits (in percent) passing the goodness-of-fit criteria. a) Exponential kernel. b) Power-law kernel.



a) b)

Figure 4.7: Comparison of theoretical and empirical densities of transformed interarrival times ξ_i. The Kullback-Leibler divergence is calculated between the theoretical density $P \sim \text{Exp}(\lambda = 1)$ and the empirical density of the residuals Q in Eq. (4.1) using the estimator developed by Wang et al. (2009).



a) b)

Figure 4.8: The empirical distribution of transformed arrival times ξ_i. a) Q-Q plot of all 60-minute look-back residuals. b) Average daily p-value of the KS test. The dashed red line represents the 5%-significance level α.

statistical tests—together with BIC-values that average across all fits.[14] We observe that with an increasing observation horizon the kernel choice becomes more and more consequential, progressively favoring the power-law variant. Indeed, on the 10-minute timescale the pass rates are almost indistinguishable, with a mild preference for the power-law kernel (based on the BIC value). This is somewhat expected, as a shorter horizon prevents the power-law kernel to leverage its long-memory property; hence, on very short timescales the use of an exponential kernel may well be justified.

The situation dramatically changes for somewhat larger observation windows, of 30-minute length, and *a fortiori* for a 60-minute observation horizon, where the power-law proves to be a superior choice. This can be deduced from the simultaneous pass for the test triad (KS $\cap$ LB $\cap$ ED), and it is further illustrated by Fig. 4.8 which features a quantile-quantile plot of the process residuals for a 60-minute time window, together with the KS p-values for both kernels (calculated as daily averages). A compounding final piece of evidence is given by the empirical distribution of the transformed time series; see Fig. 4.7—which also shows the Kullback-Leibler divergence for each kernel. Because of its consistent superiority we restrict attention to the power-law kernel.

Fig. 4.9 displays the time evolution of the endogenous and exogenous components of the process, and constitutes the Bitcoin reflexivity index for the period. It was obtained from individual fits pooled and averaged in a single point representing a 4-hour period. We observe that the level of endogeneity oscillates around the value $n = 0.8$ and consistently keeps a significant distance from criticality. This renders Bitcoin comparable to traditional FX markets which exhibit similar values of endogeneity (Lallouache and Challet, 2016; Rambaldi et al.,

[14] Wheatley et al. (2019) confirmed BIC's usefulness as an effective tool for optimal endo-exo Hawkes-model selection. The authors consider the estimation of Hawkes processes from synthetic data, with a base-rate intensity that is parametrized by log-splines of various degrees. In all considered cases, BIC is instrumental for determining the appropriate generating process.

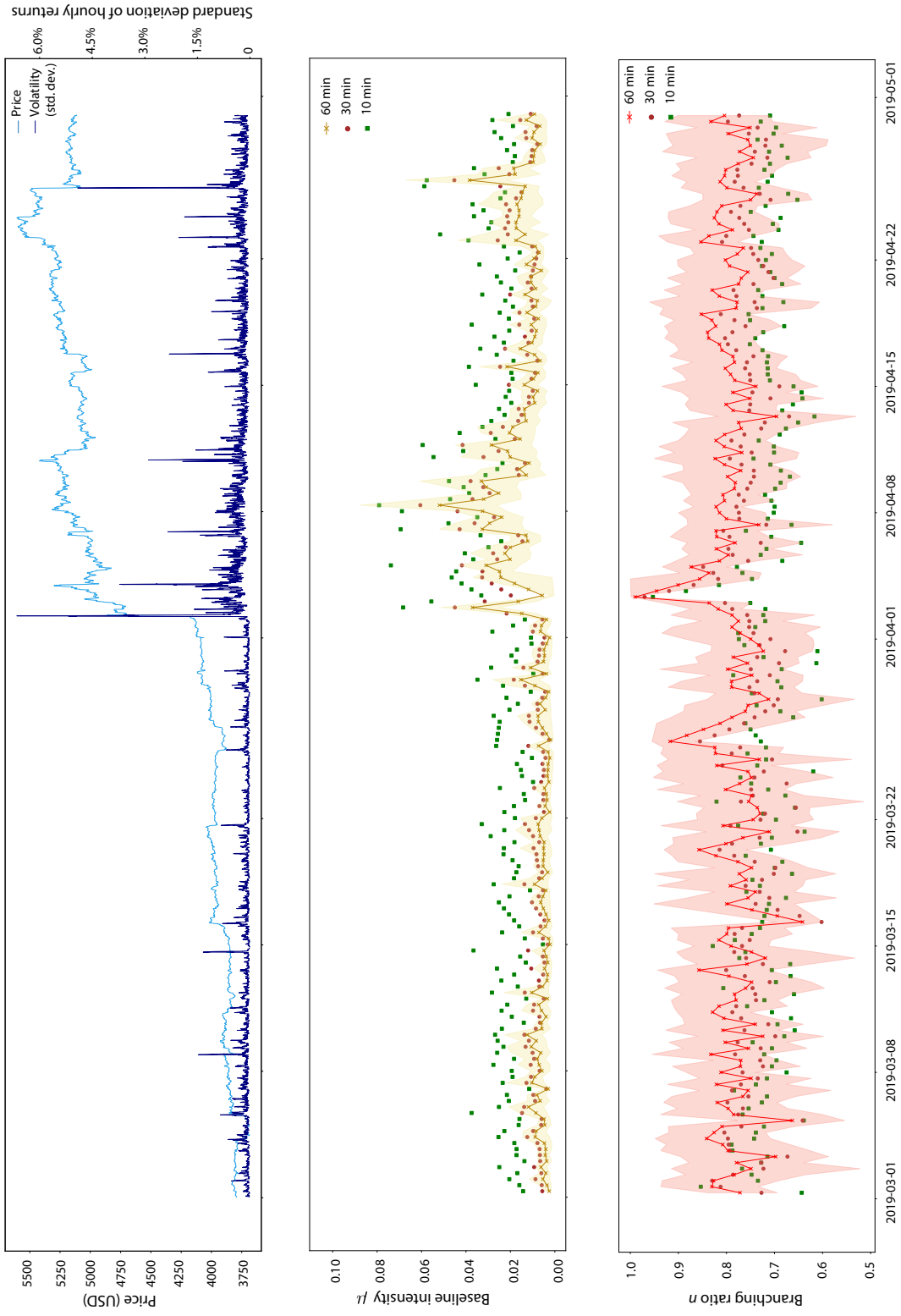


Figure 4.9: Price and volatility (in USD) in the cryptocurrency market; fluctuations of the baseline intensity (μ); Bitcoin reflexivity index n (in red, for the favored 60-minute interval), computed as a mean of 4-hour windows. The shaded areas bracket the 5%- and 95%-quantiles.

	Exponential kernel					Power-law kernel				
	KS	LB	ED	KS ∩ LB ∩ ED	Mean BIC	KS	LB	ED	KS ∩ LB ∩ ED	Mean BIC
10 min	99.32%	96.32%	98.10%	93.97%	212.71	99.79%	96.34%	99.70%	95.90%	211.12
30 min	96.44%	98.72%	90.79%	88.03%	583.35	99.19%	98.72%	98.81%	97.23%	576.03
60 min	89.19%	99.61%	74.78%	71.96%	1150.49	97.65%	99.61%	95.09%	94.19%	1131.62

Table 4.2: Fraction of acceptable fits relative to different statistical tests at the 5%-significance level. The joint pass rates for a 60-minute estimation window are given in Fig. 4.6. Look-back windows with less than 50 mid-price changes have been excluded.

2015). On the other hand, studies on other asset classes report branching ratios strikingly different. For instance, HBB find that futures on equity indices exhibit near-criticality levels of the branching ratio, while Filimonov et al. (2014) conclude that within the commodity futures market only around 60% of mid-price changes can be considered endogenous. From the perspective of market microstructure, this suggests that Bitcoin is closer to a fiat currency than gold.[15]

Measured levels of endogeneity tend to decrease with longer look-back periods, whereas base-rate estimates go up when increasing the observation horizon. Indeed, by shortening the horizon of the look-back window one tends to ignore events whose impact had not yet fully dissipated. Thus, the estimation (erroneously) credits an excessive portion of the realized intensity to the exogenous component instead of attributing it correctly to self-excitation. This confirms that even events developed on longer timescales (over tens of minutes and more) play an important role in the microstructure and therefore should not be omitted (e.g., by tightening the observation window)—at least in the absence of base-rate regime changes.

Lastly, we point out that based on the daily profile of the estimates $\hat{\mu}$ and $\hat{n}$ (Fig. 4.10), the Bitcoin exchange behaves like a true 24/7-market. Again, we emphasize the comparison with FX and equity markets where one clearly observes a “lunch lull” in the form of a U-shaped activity (that has to be accounted for in the estimation).

4.5 Conclusion

We have constructed a reflexivity index for the Bitcoin market that indicates an endogeneity of about 80%. That is, approximately four-fifth of the mid-price changes are determined within the market itself. While this value of the branching ratio is significantly lower than for equity indices (the latter being close to 1), it exceeds the reflexivity of the commodities[16] (such as gold) to which Bitcoin is often compared. The crypto-market endogeneity corresponds by and large to branching ratios found in FX markets for national currencies with which Bitcoin shares certain important traits (Barber et al., 2012; Grinberg, 2012). On the methodological

[15] Gold is often compared to Bitcoin, the former requiring physical prospecting while the latter needs virtual mining, both in need of increasing resources to extend supplies.

[16] Filimonov et al. (2014) measured the endogeneity of commodities such as sugar, wheat, or Brent crude, and found them to operate on sub-critical levels (with values between 0.4 and 0.7).

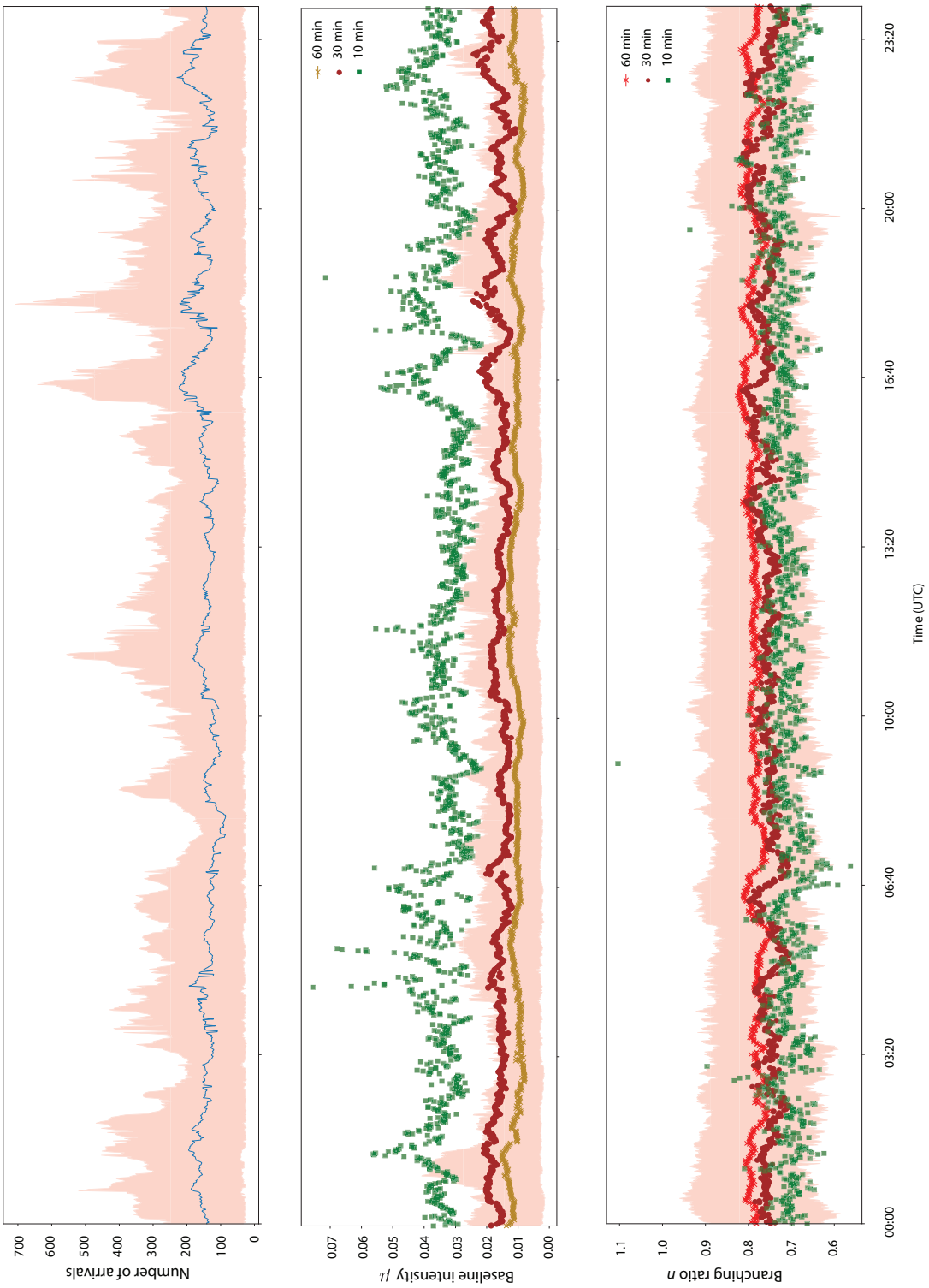


Figure 4.10: Average number of trade arrivals by time-of-day; average fluctuation of the baseline intensity (μ) by time-of-day; Bitcoin reflexivity index n by time-of-day (in red, for the favored 60-minute interval). The shaded areas represent 5%- and 95%-quantiles. [Daily profile are computed as averages over all significant fits.]

side, our study highlights the importance of determining an appropriate estimation horizon to deal with the significant nonstationarity of market activity (balancing estimation accuracy against robustness to regime changes) to reliably identify Hawkes processes with market data. Finally, our findings suggest that the generating process for Bitcoin mid-price changes features long-memory properties (i.e., a small ε in the power-law kernel), explaining the comparatively unreliable results produced by the family of exponential kernels. This is particularly evident for longer look-back windows, where the performance of the exponential kernel significantly deteriorates. Furthermore, the long-memory property of the process is observable through the impact of the window length on the estimated endo-exo levels, i.e., shorter observation horizons produce a significant upward bias in estimates of the exogenous baseline rate. To conclude, our results indicate that a Hawkes-process model with power-law kernel and optimized observation horizon(s) may well lead to excellent fits for the mid-price dynamics in Bitcoin markets.

A Supplemental material for Chapter 2

A.1 Prioritized Experience Replay

Training of the DQN architecture relies on the *experience replay buffer* which stores experience in the form of transition tuples $\tau_k = (s_k, a_k, r_k, s_{k+1})$. Rather than directly learning online and discarding the "consumed" samples thereafter, sampling from the stored transitions breaks the temporal correlations which in turn allows re-using already encountered samples in future learning. In contrast to a uniform memory buffer, prioritized sampling weights the samples so that the important ones are drawn more frequently for training. In particular, samples that produced high temporal-difference error have presumably more information to be learned from, and hence should be sampled more frequently. To this end, each transition tuple added to the memory buffer is designated with its priority $|\delta_k|$. This enriched replay buffer contains samples $(s_k, a_k, r_k, s_{k+1}, |\delta_k|)$, with priorities being updated only when the particular experience is used in a minibatch for the gradient descent. Given the absolute TD terms, we use the *proportional* method to obtain priorities p_i, i.e.,

$$p_i = |\delta_i| + \epsilon, \tag{A.1}$$

where ϵ is a small constant ensuring that any sample has some non-zero probability of being drawn.[1] Consequently, the probability distribution of the prioritized samples is given by

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \tag{A.2}$$

where α determines the level of prioritization. Indeed, for $\alpha \to 0$ we get no prioritization, i.e., the uniformly sampled experience buffer.

[1] Although functionally similar to the ϵ used for policy exploration, this constant constant is different and fixed for the whole training period.

A.1.1 Double Deep Q-Learning

Our implementation of the deep-q agent is, in fact, a double deep q architecture. That is, the agent is initialized with two networks, one target and one main with separate set of parameters $\mathbf{w}_-$ and $\mathbf{w}$. Double DQN is designed to handle the problem of overestimation of q-values. The main network is used to predict what action to take when agent encounters a new state while the target network is used to decouple the agent selection process from the construction of the new target. This way the agent is not trying to minimize loss with respect to a moving target which results in more stable learning. The TD error of the double deep q-learning defined as

$$\delta_k = r_k + \gamma \hat{q}(s_{k+1}, \arg\max_{a \in \mathscr{A}} \hat{q}(s_{k+1}, a; \mathbf{w}^-)) - \hat{q}(s_k, a_k; \mathbf{w}), \tag{A.3}$$

where the $q_{\mathbf{w}^-}$ represents the target network and $q_{\mathbf{w}}$ the main network. The target network weights are then updated from the main network every 50 learning iterations.

B Supplemental Material for Chapter 3

B.1 Appendix: Implementation Details

To ensure the exact reproducibility of Section 3.4 we now provide an exhaustive list of the hyperparameters used, followed by some practical considerations for the implementation of the DKEDPG agent discussed in the main text. The debt holders in our setting feature similar characteristics, and thus fixed repayment-process parameters. The heterogeneity in account quality is captured by the initial intensity $\lambda_0 \in \mathbb{R}_{++}$. That is, an account perceived as high quality will have a larger starting intensity in comparison with a low-quality account. Due to the nonconvex nature of likelihood estimation of the repayment-process parameters (which are observed as part of an impulse-controlled Hawkes process during ongoing collections), an efficient identification usually needs additional considerations such as a Cramér-von Mises goodness-of-fit criterion (Chehrazi and Weber, 2015) or the branching structure in an expectation-maximization algorithm (Mark and Weber, 2020).

B.1.1 Repayment Process Specification

The repayment process in an MDP environment is described in Section 3.2.1. It features a uniform distribution $\rho_0(\lambda, w)$ of initial states on the rectangular support $\mathscr{S}_0 = [\lambda_\infty, \lambda_{\max}] \times [w_{\min}, w_{\max}] \subset \mathscr{S}$. The support $\mathscr{S}_0$ also serves as an *invariant set* that contains all active states. That is, an action (or event) that would risk pushing the agent out of $\mathscr{S}_0$ is bound to receive a capped intensity increment (to ensure that the repayment intensity after the control impulse does not exceed $\lambda_{\max}$). The corresponding bounds are $w_{\min} = 1$ and $w_{\max} = 200$ (in dollars), and $\lambda_{\max} = 26.6$. The minimal balance implies that any account with $w < w_{\min}$ is considered fully collected, thus defining $[\lambda_\infty, \lambda_{\max}] \times [0, w_{\min})$ as the set of terminal states which stop the account-collection procedure. The relative repayment distribution is uniform on the support $[z_{\min}, 1]$, where $z_{\min} = 0.1$ designates the minimal relative repayment. The chosen repayment-process parameters correspond to the practical setting with a unit time period commensurate to a three-month (single-quarter) collection period. The mean-reversion constant κ is set to 0.7, and the long-run steady state is $\lambda_\infty = 0.1$. Intuitively, the mean-reversion parameter

κ determines the covariance properties of the process and can be interpreted in terms of how much memory the system retains (a larger κ increases the speed of repayment-intensity dissipation, thus decreasing the system memory). Therefore, in the absence of repayment events and account-treatment actions, the repayment intensity of an untreated account decays by $e^{-0.7\Delta t}$ after each time step. The step size $\Delta t = 0.05$ was carefully chosen as a *maximum* step size that still produces a self-exciting Hawkes process with a 99% confidence level. The sensitivity of the repayment process with respect to jumps (willingness to repay) is $\delta_{10} = 0.02$ and with respect to relative repayment sizes (ability to repay) is $\delta_{11} = 0.5$. Finally, the sensitivity of the repayment process with respect to collection effort (exerted by implementing account-treatment actions) is $\delta_2 = 1.0$, effectively normalizing the magnitude of the effort—commensurable with the repayment intensity. All admissible actions a_k are contained in the interval $[0,5]$; they are costly with a constant marginal cost of $c = 1$ (in dollars) for providing an intensity boost. The time value of money is captured by the effective (continuous-time) discount rate $\rho = 15\%$. The exact algorithm governing the MDP collections environment is sketched in Alg. 4. We note that the chosen parameters are in line with collection practice as reported by Chehrazi and Weber (2015), and the results presented in Section 3.4 are robust with respect to their particular values. Different runs were performed at different parameter configurations with qualitatively identical results.

B.1.2 Experiment Settings

Our actor implementation features a deep neural net (DNN) parametrization with two hidden layers, each spanning 64 individual neurons, as shown in Fig. B.1a. The critic network is also parameterized with a DNN. States are fed into a DNN with two hidden layers of size 16 and 32, respectively. Actions are fed into a different DNN with one hidden layer of size 32. The output of these two DNNs are combined to pass through two hidden layers of 256 neurons each; see Fig. B.1. Training is performed in batches of 512 samples using a uniform experience replay buffer at a maximum total capacity of 1,000,000 transitions. Both the critic and actor networks use an Adam optimization algorithm with a learning-rate parameter that decays linearly from 10^{-4} (resp., 2×10^{-3}) to 10^{-6} (resp., 2×10^{-6}). The penalization coefficient is 0 for the first 800 episodes of the training and 0.1 thereafter, with equal penalization for intensity and balance monotonicity (i.e., $\eta_1 = \eta_2 = 0.1$). The random exploration noise ζ is independently drawn from a Gaussian distribution with mean 0 and standard deviation 0.83. Finally, to update the target networks at each step, the permeability constant ξ is set to 0.005.

B.2 Frontier Sensitivity

The parameter vector $(\lambda_\infty, \kappa, \delta_{10}, \delta_{11}, \delta_2)$ for the repayment process in Algorithm 4 determines the location and shape of the action frontier $\mathscr{F}$. An increase of the treatment sensitivity δ_2 renders actions more effective. That is, producing a unit intensity increase becomes cheaper (compared to the base case), so the action region grows (and $\mathscr{F}$ moves up). An increase in the

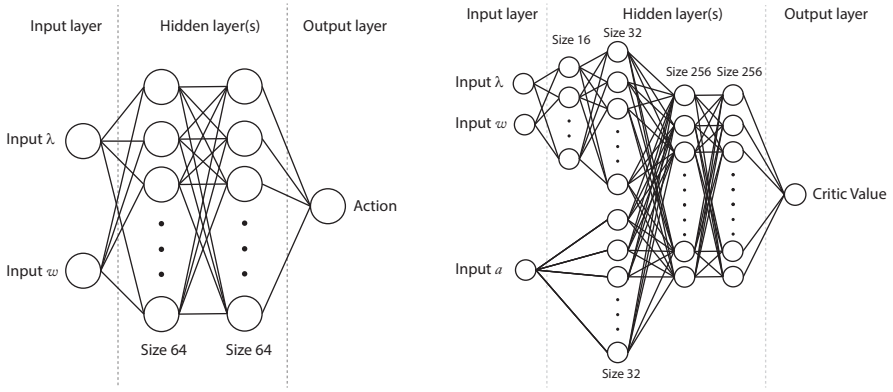


Figure B.1: a) DNN actor-critic architecture. a) Actor network b) Critic network

Algorithm 4: A discretized simulation algorithm of the repayment process from Eq. (2.1).

Result: Produces a sequence of states s_k for $k \in \{0,1,\dots,K\}$, where $w_K \leq w_{\min}$
Algorithm parameters:
$(\lambda_0, \lambda_\infty, \kappa, \delta_{10}, \delta_{11}, \delta_2)$ - process parameters, Δt - discretization step, π - policy
Initialize the current time $t = 0$, $w_k = w_0, \lambda_k = \lambda_0$
while $w_k > \epsilon_w$ **do**
  Select a according to a policy π, i.e., $a = \pi(s_k)$
  Set $\lambda_k = \lambda_k + a$
  if $\lambda_k \Delta t \geq U[0,1]$ **then**
    Draw a relative repayment z_k according to F_z
    Set $\lambda_k = \varphi(\Delta t, \lambda_k) + \delta_{10} + \delta_{11} z$
    else
      Set $z_k = 0$
      Set $\lambda_k = \varphi(\Delta t, \lambda_k)$
    end
  end
  Set $r_k = (z_k w_k - ac)$
  Set $w_k = (1 - z_k) w_k$
  Set $k = k + 1$
end

decay parameter κ implies a faster reversion towards λ_∞. Thus, maintaining the repayment intensity becomes more expensive, and $\mathscr{F}$ is pushed up for larger balances: both earlier and more forceful actions are justified by the larger expected rewards. Conversely, the treatment of low-balance accounts is delayed, for the smaller expected repayment no longer justifies the same costly collection actions. Finally, an increase of either jump sensitivity δ_{10} or repayment sensitivity δ_{11} (or both) produces larger arrival-induced intensity jumps, and therefore a higher likelihood of repayment arrivals. This decreases the need for account treatment and thus lowers the action frontier $\mathscr{F}$.

Bibliography

Abe, N., Melville, P., Pendus, C., Reddy, C. K., Jensen, D. L., Thomas, V. P. ... Miller, G. (2010). Optimizing Debt Collections Using Constrained Reinforcement Learning. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 75–84.

Adamopoulos, L. (1976). Cluster Models for Earthquakes: Regional Comparisons. *Journal of the International Association for Mathematical Geology*, 8(4):463–475.

Aït-Sahalia, Y. and Yu, J. (2008). High Frequency Market Microstructure Noise Estimates and Liquidity Measures. Technical Report, National Bureau of Economic Research, Cambridge, MA.

Ajaz, T. and Kumar, A. S. (2018). Herding in Crypto-Currency Markets. *Annals of Financial Economics*, 13(2):1850006.

Bacry, E., Jaisson, T., and Muzy, J.-F. (2016). Estimation of Slowly Decreasing Hawkes Kernels: Application to High-Frequency Order Book Dynamics. *Quantitative Finance*, 16(8):1179–1201.

Bacry, E., Mastromatteo, I., and Muzy, J.-F. (2015). Hawkes Processes in Finance. *Market Microstructure and Liquidity*, 1(1):1550005.

Barber, S., Boyen, X., Shi, E., and Uzun, E. (2012). Bitter to Better—How to Make Bitcoin a Better Currency. In: *International Conference on Financial Cryptography and Data Security*, pp. 399–414.

Baur, D. G., Hong, K., and Lee, A. D. (2018). Bitcoin: Medium of Exchange or Speculative Assets? *Journal of International Financial Markets, Institutions and Money*, 54:177–189.

Bayraktar, E. and Ludkovski, M. (2009). Sequential Tracking of a Hidden Markov Chain Using Point Process Observations. *Stochastic Processes and Their Applications*, 119(6):1792–1822.

Bertsekas, D. P. (1987). *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ.

Bertsekas, D. P. (2011). *Dynamic Programming and Optimal Control, Vol. II (Third Edition)*. Athena Scientific, Belmont, MA.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Besbes, O., Phillips, R., and Zeevi, A. (2010). Testing the Validity of a Demand Model: An Operations Perspective. *Manufacturing & Service Operations Management*, 12(1):162–183.

Black, F. (1986). Noise. *Journal of Finance*, 41(3):528–543.

Bochud, T. and Challet, D. (2007). Optimal Approximations of Power Laws with Exponentials: Application to Volatility Models with Long Memory. *Quantitative Finance*, 7(6):585–589.

Bouchaud, J.-P. (2009). The (Unfortunate) Complexity of the Economy. *Physics World*, 22-32(04):28.

Bouchaud, J.-P., Bonart, J., Donier, J., and Gould, M. (2018). *Trades, Quotes and Prices: Financial Markets under the Microscope*. Cambridge University Press, Cambridge, UK.

Bouri, E., Gupta, R., and Roubaud, D. (2019). Herding Behaviour in Cryptocurrencies. *Finance Research Letters*, 29:216–221.

Brémaud, P. and Massoulié, L. (2001). Hawkes Branching Point Processes without Ancestors. *Journal of Applied Probability*, 38(1):122–135.

Brock, W. A., Scheinkman, J. A., Dechert, W. D., and LeBaron, B. (1996). A Test for Independence Based on the Correlation Dimension. *Econometric Reviews*, 15(3):197–235.

Campbell, J. S., Givigi, S. N., and Schwartz, H. M. (2014). Multiple-Model *Q*-Learning for Stochastic Reinforcement Delays. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1611–1617.

Carvalho, D., Melo, F. S., and Santos, P. (2020). A New Convergent Variant of Q-Learning with Linear Function Approximation. *Advances in Neural Information Processing Systems*, 33:19412–19421.

Chehrazi, N., Glynn, P. W., and Weber, T. A. (2019). Dynamic Credit-Collections Optimization. *Management Science*, 65(6):2737–2769.

Chehrazi, N. and Weber, T. A. (2015). Dynamic Valuation of Delinquent Credit-Card Accounts. *Management Science*, 61(12):3077–3096.

Clark, P. J. and Evans, F. C. (1954). Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations. *Ecology*, 35(4):445–453.

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

Daley, D. J. and Vere-Jones, D. (2003). *An Introduction to the Theory of Point Processes, Vol. I: Elementary Theory and Methods*. Springer, New York, NY.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Dierckx, P. (1995). *Curve and Surface Fitting with Splines*. Oxford University Press, Oxford, UK.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. (2019). Challenges of Real-World Reinforcement Learning. *arXiv preprint arXiv:1904.12901*.

Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC Press, Boca Raton, FL.

Engel, C. (1994). Can the Markov Switching Model Forecast Exchange Rates? *Journal of International Economics*, 36(1-2):151–165.

Engle, R. F. and Russell, J. R. (1998). Autoregressive Conditional Duration: A New Model for Irregularly Spaced Transaction Data. *Econometrica*, 66(5):1127–1162.

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance*, 25(2):383–417.

Fedus, W., Ramachandran, P., Agarwal, R., Bengio, Y., Larochelle, H., Rowland, M., and Dabney, W. (2020). Revisiting Fundamentals of Experience Replay. In: *International Conference on Machine Learning*, pp. 3061–3071.

Filimonov, V., Bicchetti, D., Maystre, N., and Sornette, D. (2014). Quantification of the High Level of Endogeneity and of Structural Regime Shifts in Commodity Markets. *Journal of International Money and Finance*, 42:174–192.

Filimonov, V. and Sornette, D. (2012). Quantifying Reflexivity in Financial Markets: Toward a Prediction of Flash Crashes. *Physical Review E*, 85(5):056108.

Filimonov, V. and Sornette, D. (2015). Apparent Criticality and Calibration Issues in the Hawkes Self-Excited Point Process Model: Application to High-Frequency Financial Data. *Quantitative Finance*, 15(8):1293–1314.

Gelfand, I. M. and Fomin, S. V. (1963). *Calculus of Variations*. Prentice-Hall, Englewood Cliffs, NJ.

Ghassemi, M., Naumann, T., Schulam, P., Beam, A. L., Chen, I. Y., and Ranganath, R. (2020). A Review of Challenges and Opportunities in Machine Learning for Health. In: *AMIA Summits on Translational Science Proceedings*, pp. 191–200.

Grinberg, R. (2012). Bitcoin: An Innovative Alternative Digital Currency. *Hastings Science and Technology Law Journal*, 4(1):159–208.

Grossman, S. J. (1989). *The Informational Role of Prices*. MIT Press, Cambridge, MA.

Guo, X., Hu, A., Xu, R., and Zhang, J. (2018). Consistency and Computation of Regularized MLEs for Multivariate Hawkes Processes. *arXiv preprint arXiv:1810.02955*.

Gupta, A., Shukla, N., Marla, L., Kolbeinsson, A., and Yellepeddi, K. (2019). How to Incorporate Monotonicity in Deep Networks While Preserving Flexibility? *arXiv preprint arXiv:1909.10662*.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In: *International Conference on Machine Learning*, pp. 1861–1870.

Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52.

Hardiman, S. J., Bercot, N., and Bouchaud, J.-P. (2013). Critical Reflexivity in Financial Markets: A Hawkes Process Analysis. *European Physical Journal B*, 86(10):442.

Hardiman, S. J. and Bouchaud, J.-P. (2014). Branching-Ratio Approximation for the Self-Exciting Hawkes Process. *Physical Review E*, 90(6):062807.

Hawkes, A. and Adamopoulos, L. (1973). Cluster Models for Earthquakes-Regional Comparisons. *Bulletin of the International Statistical Institute*, 45(3):454–461.

Hawkes, A. G. (1971a). Point Spectra of Some Mutually Exciting Point Processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443.

Hawkes, A. G. (1971b). Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1):83–90.

Hawkes, A. G. and Oakes, D. (1974). A Cluster Process Representation of a Self-Exciting Process. *Journal of Applied Probability*, 11(3):493–503.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W. ... Silver, D. (2017). Rainbow: Combining Improvements in Deep Reinforcement Learning. *arXiv preprint arXiv:1710.02298*.

Hong, H., Kubik, J. D., and Stein, J. C. (2005). Thy Neighbor's Portfolio: Word-of-Mouth Effects in the Holdings and Trades of Money Managers. *Journal of Finance*, 60(6):2801–2824.

Huber, P. J. (1992). Robust Estimation of a Location Parameter. In: Kotz, S. and Johnson, N. L. (Eds.). *Breakthroughs in Statistics*. Springer, New York, NY, pp. 492–518.

Jiang, Y., Nie, H., and Ruan, W. (2018). Time-Varying Long-Term Memory in Bitcoin Market. *Finance Research Letters*, 25:280–284.

Keynes, J. M. (1936). *The General Theory of Employment, Interest and Money*. Macmillan, London, UK.

Kristoufek, L. (2015). What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. *PloS One*, 10(4):e0123923.

Kristoufek, L. (2018). On Bitcoin Markets (In)Efficiency and Its Evolution. *Physica A: Statistical Mechanics and Its Applications*, 503:257–262.

Lallouache, M. and Challet, D. (2016). The Limits of Statistical Significance of Hawkes Processes Fitted to Financial Data. *Quantitative Finance*, 16(1):1–11.

LeRoy, S. F. and Porter, R. D. (1981). The Present-value Relation: Tests Based on Implied Variance Bounds. *Econometrica*, 49(3):421–436.

Lewis, M. and Baker, D. (2014). *Flash Boys: A Wall Street Revolt*. Norton, New York, NY.

Liebman, L. H. (1972). A Markov Decision Model for Selecting Optimal Credit Control Policies. *Management Science*, 18(10):B–519–B–525.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y. ... Wierstra, D. (2015). Continuous Control with Deep Reinforcement Learning. *arXiv preprint arXiv:1509.02971*.

Liu, R. and Zou, J. (2018). The Effects of Memory Replay in Reinforcement Learning. In: *2018 56th Annual Allerton Conference on Communication, Control, and Computing*, pp. 478–485.

Ljung, G. M. and Box, G. E. (1978). On a Measure of Lack of Fit in Time Series Models. *Biometrika*, 65(2):297–303.

Lux, T. (1995). Herd Behaviour, Bubbles and Crashes. *Economic Journal*, 105(431):881–896.

Mark, M., Chehrazi, N., Liu, H., and Weber, T. A. (2021). Optimal Recovery of Unsecured Debt via Interpretable Reinforcement Learning. Working Paper, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Mark, M., Chehrazi, N., and Weber, T. A. (2020a). Reinforcement-Learning Approach to Credit Collections. Working Paper, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Mark, M., Sila, J., and Weber, T. A. (2020b). Quantifying Endogeneity of Cryptocurrency Markets. *European Journal of Finance*. DOI: 10.1080/1351847X.2020.1791925.

Mark, M. and Weber, T. A. (2020). Robust Identification of Controlled Hawkes Processes. *Physical Review E*, 101(4):043305.

Melo, F. S. (2001). Convergence of Q-Learning: A Simple Proof. Technical Report, Institute of Systems and Robotics, Lisbon, Portugal.

Melo, F. S. and Ribeiro, M. I. (2007). Convergence of Q-Learning with Linear Function Approximation. In: *2007 European Control Conference*, pp. 2671–2678.

Miller, G., Weatherwax, M., Gardinier, T., Abe, N., Melville, P., Pendus, C. ... Cooley, B. (2012). Tax Collections Optimization for New York State. *Interfaces*, 42(1):74–84.

Minka, T. (1998). Expectation-Maximization as Lower Bound Maximization. Working Paper, Pennsylvania State University, University Park, PA.

Mitchner, M. and Peterson, R. P. (1957). An Operations-Research Study of the Collection of Defaulted Loans. *Operations Research*, 5(4):522–545.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*.

Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., and Tita, G. E. (2011). Self-Exciting Point Process Modeling of Crime. *Journal of the American Statistical Association*, 106(493):100–108.

Møller, J. and Rasmussen, J. G. (2006). Approximate Simulation of Hawkes Processes. *Methodology and Computing in Applied Probability*, 8(1):53–64.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. White Paper.

Ogata, Y. (1978). The Asymptotic Behaviour of Maximum Likelihood Estimators for Stationary Point Processes. *Annals of the Institute of Statistical Mathematics*, 30(2):243–261.

Ogata, Y. (1981). On Lewis' Simulation Method for Point Processes. *IEEE Transactions on Information Theory*, 27(1):23–31.

Ogata, Y. (1988). Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes. *Journal of the American Statistical Association*, 83(401):9–27.

Ogata, Y. (1993). Space-Time Modelling of Earthquake Occurrences. *Bulletin of the International Statistical Institute*, 55(2):249–250.

Ogata, Y. (1998). Space-Time Point-Process Models for Earthquake Occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402.

Ogata, Y. and Akaike, H. (1982). On Linear Intensity Models for Mixed Doubly Stochastic Poisson and Self-Exciting Point Processes. In: Parzen, E., Tanabe, K., and Kitagawa, G. (Eds.). *Selected Papers of Hirotugu Akaike*. Springer, New York, NY, pp. 269–274.

Ogata, Y. and Zhuang, J. (2006). Space–Time ETAS Models and an Improved Extension. *Tectonophysics*, 413(1-2):13–23.

Ozaki, T. (1979). Maximum Likelihood Estimation of Hawkes' Self-Exciting Point Processes. *Annals of the Institute of Statistical Mathematics*, 31(1):145–155.

Peskir, G. and Shiryaev, A. N. (2002). Solving the Poisson Disorder Problem. In: *Advances in Finance and Stochastics*. Springer, New York, NY, pp. 295–312.

Piano, S. L. (2020). Ethical Principles in Machine Learning and Artificial Intelligence: Cases from the Field and Possible Ways Forward. *Humanities and Social Sciences Communications*, 7(1):1–7.

Rambaldi, M., Pennesi, P., and Lillo, F. (2015). Modeling Foreign Exchange Market Activity around Macroeconomic News: Hawkes-Process Approach. *Physical Review E*, 91(1):012819.

Reinhard, C. M. and Rogoff, K. S. (2009). *This Time is Different: Eight Centuries of Financial Folly*. Princeton University Press, Princeton, NJ.

Rubin, I. (1972). Regular Point Processes and Their Detection. *IEEE Transactions on Information Theory*, 18(5):547–557.

Rudin, W. et al. (1976). *Principles of Mathematical Analysis (Third Edition)*. McGraw-Hill, New York, NY.

Shiller, R. J. (1981). Do Stock Prices Move Too Much to Be Justified by Subsequent Changes in Dividends? *American Economic Review*, 71(3):421–436.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic Policy Gradient Algorithms. In: *International Conference on Machine Learning*, pp. 387–395.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A. ... Hassabis, D. (2017). Mastering the Game of Go without Human Knowledge. *Nature*, 550(7676):354–359.

Soros, G. (1994). *The Alchemy of Finance: Reading the Mind of the Market*. Wiley, New York, NY.

Stokey, N. L. (2008). *The Economics of Inaction*. Princeton University Press, Princeton, NJ.

Sutton, R. S. (1988). Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, 3(1):9–44.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction (Second Edition)*. MIT Press, Cambridge, MA.

Truccolo, W. (2016). From Point Process Observations to Collective Neural Dynamics: Nonlinear Hawkes Process GLMs, Low-Dimensional Dynamics and Coarse Graining. *Journal of Physiology (Paris)*, 110(4):336–347.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N. (2005). A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects. *Journal of Neurophysiology*, 93(2):1074–1089.

Tsitsiklis, J. N. and Van Roy, B. (1997). An Analysis of Temporal-Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690.

Upadhyay, U., De, A., and Rodriguez, M. G. (2018). Deep Reinforcement Learning of Marked Temporal Point Processes. In: *Advances in Neural Information Processing Systems*, pp. 3168–3178.

Urquhart, A. (2016). The Inefficiency of Bitcoin. *Economics Letters*, 148:80–82.

Valera, I. and Gomez-Rodriguez, M. (2015). Modeling Adoption and Usage of Competing Products. In: *2015 IEEE International Conference on Data Mining*, pp. 409–418.

Veen, A. and Schoenberg, F. P. (2008). Estimation of Space–Time Branching Process Models in Seismology Using an EM–Type Algorithm. *Journal of the American Statistical Association*, 103(482):614–624.

Vere-Jones, D. (1975). Stochastic Models for Earthquake Sequences. *Geophysical Journal International*, 42(2):811–826.

Vidal-Tomás, D., Ibáñez, A. M., and Farinós, J. E. (2019). Weak Efficiency of the Cryptocurrency Market: A Market Portfolio Approach. *Applied Economics Letters*, 26(19):1627–1633.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J. ... Silver, D. (2019). Grandmaster Level in Starcraft II Using Multi-Agent Reinforcement Learning. *Nature*, 575(7782):350–354.

Wang, Q., Kulkarni, S. R., and Verdú, S. (2009). Divergence Estimation for Multidimensional Densities via k-Nearest-Neighbor Distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405.

Watkins, C. J. and Dayan, P. (1992). Q-Learning. *Machine Learning*, 8(3–4):279–292.

Wehrli, A., Wheatley, S., and Sornette, D. (2021). Scale-, Time-and Asset-Dependence of Hawkes Process Estimates on High Frequency Price Changes. *Quantitative Finance*, 21(5):729–752.

Wheatley, S., Wehrli, A., and Sornette, D. (2019). The Endo–Exo Problem in High Frequency Financial Price Fluctuations and Rejecting Criticality. *Quantitative Finance*, 19(7):1165–1178.

Xu, L., Duan, J. A., and Whinston, A. (2014). Path to Purchase: A Mutually Exciting Point Process Model for Online Advertising and Conversion. *Management Science*, 60(6):1392–1412.

You, S., Ding, D., Canini, K., Pfeifer, J., and Gupta, M. (2017). Deep Lattice Networks and Partial Monotonic Functions. In: *Advances in Neural Information Processing Systems*, pp. 2981–2989.

Zhou, K., Zha, H., and Song, L. (2013). Learning Triggering Kernels for Multi-Dimensional Hawkes Processes. In: *International Conference on Machine Learning*, pp. 1301–1309.

Zhuang, J., Ogata, Y., and Vere-Jones, D. (2002). Stochastic Declustering of Space-Time Earthquake Occurrences. *Journal of the American Statistical Association*, 97(458):369–380.

Michael Mark

Personal Details

Address:	81 Oberstrasse, 9000 St. Gallen, Switzerland
Telephone:	+41 78 734 78 16
Email:	michael.mark@epfl.ch
Social:	Github : majkee15, LinkedIn

Education

September 2017 - January 2022 — **École Polytechnique Fédérale de Lausanne**, Lausanne, CH
Ph.D., Operations, Economics and Strategy

Ph.D. Thesis: Self-Exciting Point Processes: Identification and Control
(Advisor: Prof. Thomas A. Weber)

September 2015 - September 2016 — **University of Leicester**, Leicester, UK
MSc., Financial Mathematics and Computation
Graduated with *Distinction*

MS Thesis: Pricing of Exotics under Stochastic Volatility
(Advisors: Prof. Jeremy Levesley and Prof. Bogdan Grechuk)

August 2013 - June 2015 — **La Sorbonne, Université Pierre et Marie Curie**, Paris, FR
Licence (Undergraduate) Mathématiques,
Mention très bien, grade: 16.68/20

Professional Experience

November 2016 - July 2017 — **Ernst & Young,** London, UK
Associate Consultant - Quantitative Risk Analyst
Key engagements included:

- Development, validation, and implementation of a VaR based risk engine, responsible for the interest-rate products
- Validation of internal credit risk models (PD/LGD/EAD) for UK financial institutions/building societies
- Derivatives valuation (rates, equity, fixed income) for the assurance unit
- Stress testing: PRA and EBA stress tests

July 2014 - September 2014 — **Cyrrus a.s.,** Prague, CZ
Summer Analyst - Internship
Responsible for:

- Compilation and delivery of various client material (slide decks for new products, portfolio performance reports)
- Inhouse analytics research (DCF models, risk metric evaluations, composition and computation of internal risk factors)

Awards & Recognition

- **1st place, Best Student Paper on Finance Competition**, INFORMS Annual Meeting, 2020
- **1st place, Operations Research Challenge**, INFORMS Annual Meeting, Phoenix, AZ, 2018
- **Best Student Performance Award**, University of Leicester, UK, 2016
- **Academic Grant for UCM Modelling Week**, Universidad Complutense de Madrid, ESP, 2016
- **1st place, Deloitte Applied Analytics Challenge**, Prague, CZE, 2016

Skills

Python	6+ years of experience. Expert knowledge of modern machine learning and data science libraries (*Scikit-learn, Pandas, Numpy, Tensorflow 2.0*, and many others), author of several packages, and an open-source contributor. Experienced with linking low-level C/C++ code with high-level Python via *Cython*. **Sample projects:** AWS-based high-frequency database for quotes and trades data built on web-sockets, complete trade execution system running concurrently on several cryptocurrency exchanges, reinforcement-learning academic research with multiple publications, implementation of credit risk and market risk models.
C++	5+ years of experience. Used for writing highly performant code that interfaces directly with Python. Good knowledge of *Boost* and *QuantLib*. **Sample projects:** Stochastic volatility Heston pricer for path-dependent options, Hawkes model simulation, and inference via MLE/EM method for various kernels, derivatives pricing.
R	5+ years of experience. Mainly used for time series analysis and data wrangling.
Others	C#, Julia, Mathematica, Matlab, Rust, SQL, VBA.

Publications

1. **Mark, M.**, Sila, J., and Weber, T. A. (2020). Quantifying Endogeneity of Cryptocurrency Markets. *European Journal of Finance*, Forthcoming. DOI: 10.1080/1351847X.2020.1791925. [Open Access]
2. **Mark, M.** and Weber, T. A. (2020). Robust Identification of Controlled Hawkes Processes. *Physical Review E*, 101(4):043305. DOI: 10.1103/PhysRevE.101.043305. [Open Access]
3. Seda, P., **Mark, M.**, Su, K.-W., Seda, M., Hosek, J., and Leu, J.-S. (2019). The Minimization of Public Facilities with Enhanced Genetic Algorithms Using War Elimination. *IEEE Access*, 7:9395–9405. DOI: 10.1109/ACCESS.2019.2891424. [Open Access]
4. **Mark, M.**, Chehrazi, N., and Weber, T. A. (2020). Reinforcement-Learning Approach to Credit Collections. Working Paper, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.
5. **Mark, M.**, Chehrazi, N., Liu, H., and Weber, T. A. (2021). Optimal Recovery of Unsecured Debt via Interpretable Reinforcement Learning. Working Paper, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Languages

English:	Fluent	(CAE C1, IELTS 7.5)
French:	Fluent	Undergraduate degree taught in French
German:	Intermediate	
Czech:	Native	
Slovak:	Fluent	

Extracurricular activities

Beach volleyball:	2010 - 2013	Czech National Draft (U20-U23)
Volleyball:	2013 - 2014	League National Française, Fontenay aux Roses, Paris
	2013 - 2014	French Varsity, UPMC, Paris
	2016 - 2017	English National Volleyball League, Lynx, London
	2017 - 2019	Swiss Volleyball League, LUC, Lausanne