Machine Learning with Applications 8 (2022) 100280

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Optimal recovery of unsecured debt via interpretable reinforcement learning

Michael Mark [a,*], Naveed Chehrazi [b], Huanxi Liu [a], Thomas A. Weber [a]

[a] École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland
[b] Olin Business School, Washington University in St. Louis, St. Louis, MO 63130, USA

ARTICLE INFO

Keywords:
Reinforcement learning
Interpretable machine learning
Deterministic policy gradient
Monotonicity constrained learning
Debt recovery
Control of Hawkes processes

ABSTRACT

This paper addresses the issue of interpretability and auditability of reinforcement-learning agents employed in the recovery of unsecured consumer debt. To this end, we develop a deterministic policy-gradient method that allows for a natural integration of domain expertise into the learning procedure so as to encourage learning of consistent, and thus interpretable, policies. Domain knowledge can often be expressed 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 which integrates domain expertise into the learning. Our formulation overcomes the challenge of learning interpretable policies by constraining the search to policies satisfying structural-consistency properties. The resulting state-feedback control laws can be readily understood and implemented by human decision makers. This new domain-knowledge enhanced learning approach is applied to the problem of optimal debt recovery which features a controlled Hawkes process and an asynchronous action–feedback relationship.

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 superhuman 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 information 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 (DM). 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 (i.e., minorities) for which the available data are relatively sparse. Therefore, the notion of interpretability and systemic consistency is closely connected 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 (Rudin et al., 2021). For instance, You, Ding, Canini, Pfeifer, and Gupta (2017) propose a deep lattice framework (as a counterpart to neural nets) to learn flexible monotonic functions, and Gupta, Shukla, Marla, Kolbeinsson, and Yellepeddi (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 (i.e., systemically consistent), whence interpretable by human decision makers. 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

* Corresponding author.
E-mail addresses: michael.mark@epfl.ch (M. Mark), naveed.chehrazi@wustl.edu (N. Chehrazi), huanxiliu99@gmail.com (H. Liu), thomas.weber@epfl.ch (T.A. Weber).

https://doi.org/10.1016/j.mlwa.2022.100280
Received 7 December 2021; Accepted 23 February 2022
Available online 2 March 2022

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 (DK) directly into the learning process to obtain interpretable policies. By design, we narrow our focus to quantifiable domain knowledge which can then be embedded into the learning. 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 of desirable and explainable policies with less variance across runs. We showcase the relevance of our approach in the context of optimal debt recovery, a practically relevant stochastic control problem which features a self-exciting (Hawkes) repayment process and an asynchronous learning feedback.

2. Background

2.1. Preliminaries

We study a specific type of reinforcement-learning problem, 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 debt recovery 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 debt-recovery 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 debt-recovery problem as a Markov decision process (MDP) with a state space $\mathcal{S}$, an action space $\mathcal{A}$, transition probabilities $P(s_{k+1}, s_k, a_k)$, an initial state distribution ρ_0 (on $\mathcal{S}$), a reward function $\mathcal{R} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$, 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),

$$d\lambda(t) = \underbrace{\kappa(\lambda_\infty - \lambda(t))dt}_{\text{mean-reversion}} + \underbrace{\delta_1^\top dJ(t)}_{\text{self-excitation}} + \underbrace{dA(t)}_{\text{collection strategy}}, \quad t \geq 0. \tag{1}$$

This mean-reverting SDE describes the dynamics of the repayment arrival rate (intensity) for an account placed in collections at time $t = 0$ with (given) initial intensity $\lambda(0) = \lambda$. Eq. (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. (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. (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 autocovariance properties of the process, determines how much "memory" the system retains. Unlike the intensity dynamics in Eq. (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, so $w(\tau_i) = (1 - z_i)w(\tau_i^-)$, where $z_i = b_i/w_{i-1}$ for $i \geq 1$. Hence,

$$w(t) = w(\tau_i), \qquad \tau_i \leq t < \tau_{i+1}. \tag{2}$$

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

$$\lambda\left(t' | \lambda(t)\right) = \varphi\left(t', \lambda(t)\right) = \lambda_\infty + (\lambda(t) - \lambda_\infty)e^{-\kappa t'}, \quad t' \geq t, \tag{3}$$

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 debt-recovery problem into a reinforcement-learning framework, the continuous-time Markovian dynamics in Eqs. (1) and (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

$$\begin{aligned}&\mathbb{P}[N\left((k+1)\Delta t\right) - N(k\Delta t) = n | \mathcal{H}_{k\Delta t}] \\ &= \begin{cases} 1 - \left(\lambda(k\Delta t) + a_{k\Delta t}\right)\Delta t + o(\Delta t), & n = 0, \\ \left(\lambda(k\Delta t) + a_{k\Delta t}\right)\Delta t + o(\Delta t), & n = 1, \\ o(\Delta t), & n \geq 2. \end{cases}\end{aligned} \tag{4}$$

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

$$\lambda(k\Delta t) = \varphi\left(\Delta t, \lambda\left((k-1)\Delta t\right) + a_{(k-1)\Delta t}\right) + (\delta_{10} + \delta_{11} z_{k-1})\mathbb{1}_{\{N(k\Delta t) - N((k-1)\Delta t) \neq 0\}}, \tag{5}$$

with $\lambda(0) = \lambda_0$, where we are allowed to use Eq. (3), since no discrete event will take place on the interval $((k-1)\Delta t, k\Delta t)$. 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

$$w(k\Delta t) = (1 - z_{k-1})w\left((k-1)\Delta t\right)\mathbb{1}_{\{N(k\Delta t) - N((k-1)\Delta t) \neq 0\}}, \quad k \geq 0, \tag{6}$$

with $w(0) = w_0$ and $z_{-1} = 0$. Eqs. (4)–(6) fully describe the discrete-time dynamics of the debt-recovery process. To simplify the notation, in what follows we denote the tuple $\left(\lambda(k\Delta t), w(k\Delta t), a_{k\Delta t}\right)$ 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 $S \subsetneq \mathbb{R}_+^2$. This last step turns the discrete-time, continuous-space Markov dynamics of Eqs. (4)–(6) into a discrete-time finite-state Markov chain, but otherwise this

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

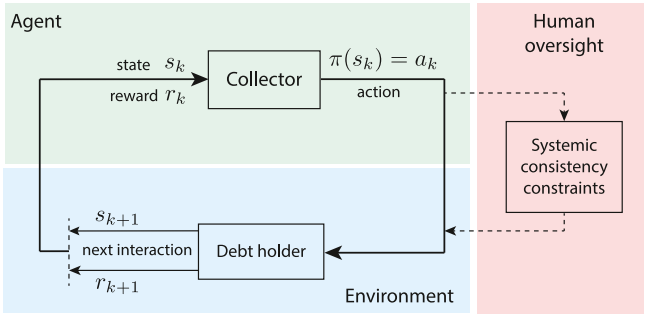


Fig. 1. Illustration of a Markov decision process with human oversight. At the end of the learning, the learned policy $\pi(\cdot)$ is subjected to a human DM for validation. Inconsistent and non-interpretable policies undermine the DM's confidence in the model and thus are discarded.

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. 2.

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. (4)–(6). The agent, following a policy $\pi : \mathcal{S} \to \mathbb{R}_+$ that prescribes his action for a given state, repeatedly interacts with the environment (see Fig. 1). At each (discretized) time step $k \geq 0$, the agent observes the state $s_k = (\lambda_k, w_k) \in \mathcal{S}$, selects an action $\pi(s_k) = a_k \in \mathcal{A} = \mathbb{R}_+$ 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

$$r_k \triangleq \mathcal{R}(s_k, a_k, s_{k+1}) = \begin{cases} \gamma z_k w_k - c a_k, & \text{repayment in } [k\Delta t, (k+1)\Delta t], \\ -c a_k, & \text{otherwise,} \end{cases} \tag{7}$$

where $\gamma \in (0, 1)$. The agent's goal is to find a policy π that maximizes expected net collections,

$$v_\pi(s_0) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^{k+1} w_k z_k - c \sum_{k=0}^{\infty} \gamma^k a_k \middle| \mathcal{H}_0 \right], \tag{8}$$

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

Remark 1. Chehrazi, Glynn, and Weber (2019) proposed and solved a variation of the debt-recovery problem outlined above. Specifically, 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 actively pursued, for instance, until an agreement for a repayment plan is reached. The reason for examining an analytically solved problem is threefold. Firstly, the optimal solution provides a clear performance benchmark as well as a neighborhood comparison in the policy space. Secondly, Eq. (1) represents the fundamental debt-recovery dynamics which invites further extensions that capture more nuanced elements of the repayment behavior. For example, while nonlinear collection actions and debtor responses (e.g., actions with diminishing impact, fixed costs, state-dependent repayment distribution) are perhaps more realistic, they tend to render the problem intractable from an analytical standpoint. Finally, the stochastic differential equation in Eq. (1) comprises a large class of asynchronous control problems with self-exciting intensity, so that this model can be viewed as a template for similar problems, which arise in the accumulation of innovation, dynamic advertising, and the control of other stochastic arrival processes, for example, in social networks.

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

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 $\pi_\theta : \mathcal{S} \to \mathcal{A}$ with parameter vector $\theta \in \mathbb{R}^{d_1}$ of dimension $d_1 \ll |\mathcal{S}|$. Let $\rho_{\pi_\theta}(s') = \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \rho_0(s) P_t(s, s'; \pi_\theta) ds$ be the distribution of the discounted state visits, where $P_t(s, s'; \pi_\theta)$ denotes the probability of going from s to s' in t steps under a policy π_θ, i.e., $\mathbb{P}(s_{k+t} = s' | s_k = s, \pi_\theta)$.[3] We define an optimal policy $\pi_\theta^\star$ such that $\pi_\theta^\star \in \arg\max_\theta J(\pi_\theta)$, where

$$J(\pi_\theta) \triangleq \mathbb{E}_{s_0 \sim \rho_0}[v_{\pi_\theta}(s_0)] = \int_{\mathcal{S}} \rho_{\pi_\theta}(s) r(s, \pi_\theta(s)) ds = \mathbb{E}_{s \sim \rho_{\pi_\theta}} \left[r(s, \pi_\theta(s)) \right], \tag{9}$$

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

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= \int_{\mathcal{S}} \rho_{\pi_\theta}(s) \nabla_a q_{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} \nabla_\theta \pi_\theta(s) \, ds \\ &= \mathbb{E}_{s \sim \rho_{\pi_\theta}} \left[\nabla_a q_{\pi_\theta}(s, a) \nabla_\theta \pi_\theta(s)|_{a=\pi_\theta(s)} \right], \end{aligned}$$

where $q_{\pi_\theta}(s, a) = r(s, a) + \gamma \int_{\mathcal{S}} P_1(s, s', \pi_\theta) v_{\pi_\theta}(s') ds'$ is the q-function associated with Eq. (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 using a parameter vector $\mathbf{w} \in \mathbb{R}^{d_2}$, i.e., $q_{\pi_\theta}(s, a) = \hat{q}_{\pi_\theta}(s, a; \mathbf{w})$, and is learned by sequentially minimizing a loss of the form

$$\mathcal{L}(\mathbf{w}_l) = \mathbb{E}_{(s_k, a_k, r_k, s_{k+1}) \sim D} \left[\frac{1}{2} \left(r_k + \gamma \hat{q}_{\pi_\theta}(s_{k+1}, \pi(s_{k+1}); \mathbf{w}_{l-1}) - \hat{q}_{\pi_\theta}(s_k, a_k; \mathbf{w}_l) \right)^2 \right], \tag{10}$$

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 $\pi_\theta(\cdot)$ then ascends his payoff in the direction of the gradient of the objective function,

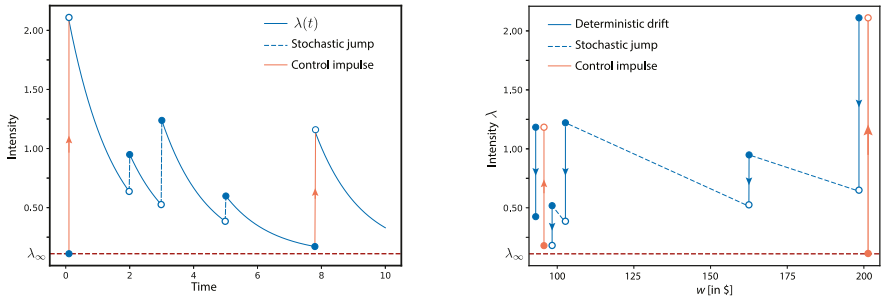
$$J_\beta^{(l)}(\pi_\theta) = \int_{\mathcal{S}} \rho_\beta(s) v_{\pi_\theta}(s) ds = \mathbb{E}_{s \sim \rho_\beta(\cdot)} \left[\hat{q}_{\pi_\theta}(s, \pi_\theta(s); \mathbf{w}_l) \right], \tag{11}$$

where $\beta : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is an arbitrary, possibly stochastic, exploration distribution (*behavioral* policy) such that $\int_{\mathcal{A}} \beta(s, a) da = 1$ for all $s \in \mathcal{S}$. 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 the fact 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. Incorporating domain knowledge

For the optimal debt-recovery problem, the interpretability of a given policy π to a human collector is closely linked to the structure

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



(a) Self-exciting intensity (t, $\lambda(t)$) (b) Account state $(\lambda, w) \in S$

Fig. 2. Controlled state-transition dynamics with two action-induced jumps.

of the action set in the state space. This structure must 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,

$$\left\{ \begin{array}{rcl} w' \leq w & \Rightarrow & \pi_{\boldsymbol{\theta}}(\lambda, w') \leq \pi_{\boldsymbol{\theta}}(\lambda, w) \\ \lambda \leq \lambda' & \Rightarrow & \pi_{\boldsymbol{\theta}}(\lambda', w) \leq \pi_{\boldsymbol{\theta}}(\lambda, w) \end{array} \right\}. \tag{12}$$

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 in a higher balance state; similarly, an account in a 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 debt-recovery problem in continuous time. The monotonicity constraints in Eq. (12) can be included in the learning by means of a barrier regularization term,

$$H(\pi_{\boldsymbol{\theta}}(\lambda, w)) = \eta_1 \max\left\{0, \frac{\partial \pi_{\boldsymbol{\theta}}(\lambda, w)}{\partial \lambda}\right\} + \eta_2 \max\left\{0, -\frac{\partial \pi_{\boldsymbol{\theta}}(\lambda, w)}{\partial w}\right\}, \tag{13}$$

where η_1 and η_2 are positive constants. Similar to a maximum-entropy policy-gradient framework where a regularizer encourages learning of explorative policies (Haarnoja, Zhou, Abbeel, & Levine, 2018), we add the regularizer to the off-policy performance metric in Eq. (11), so

$$\hat{J}(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{s\sim\rho_\beta(\cdot)}\left[q_{\pi_{\boldsymbol{\theta}}}(s, \pi_{\boldsymbol{\theta}}(s)) - H(\pi_{\boldsymbol{\theta}}(s))\right]; \tag{14}$$

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

$$\nabla_{\boldsymbol{\theta}}\hat{J}(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{s\sim\rho_\beta(\cdot)}\left[\nabla_a \hat{q}_{\pi_{\boldsymbol{\theta}}}(s, a)|_{a=\pi_{\boldsymbol{\theta}}(s)} \nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(s) - \nabla_{\boldsymbol{\theta}} H(\pi_{\boldsymbol{\theta}}(s))\right]. \tag{15}$$

The intuition behind the shape regularizer (which can easily be augmented to also contain higher-order monotonicities, e.g., to capture the concavity of the action frontier with respect to w) is to reject critical points in the policy space that yield locally non-interpretable policies (i.e., violating Eq. (12)) 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 (deterministic) Policy Gradient (DKEPG), see Alg. 1.

Remark 2. The applicability of monotonicity, convexity, and other shape constraints spans far beyond our particular application in debt recovery and is pervasive in operations research (Chehrazi & Weber, 2010). Thus, our approach may be used for a sizable class of stochastic accumulation problems which tend to exhibit the aforementioned structural properties of value function and optimal policy. Alternatively, it can improve model-interpretability (consistency) in decision-making applications where the agent has an intuitive interpretation of the state variables in terms of monotone comparative statics; the latter guide an intuitive understanding of system inputs (states) and their impact on the value function and/or policy prescriptions. To illustrate the breadth of applications, consider the following three MDPs:

(i) An agent obtains utility $u(c_t)$ when consuming wealth c_t at time $t \in \{0, 1, \ldots\}$. The agent receives a random (nonnegative) income y_t at time t, and we assume that y_t is a Markov process described by a stochastic transition matrix $P(y_t, y')$. Let w_t denote the total wealth of the agent at time t, so $w_{t+1} = w_t + y_t - c_t$ for all $t \geq 0$, with $w_0 \geq 0$ a given initial wealth level. We seek an optimal consumption $c_t^\star$ for the value function as a solution to the Bellman equation

$$v(w_t, y_t) = \max_{c\in[0, w_t+y_t]}\left\{u(c) + \gamma \sum_{y'} v(w_t + y_t - c, y')P(y_t, y')\right\},$$

for all $t \geq 0$. Clearly, under the assumption of a (continuous) increasing utility the value function is *increasing* in wealth. Our approach enables the DM to readily embed this specific domain knowledge into the learning algorithm.

(ii) Transportation platforms, such as Uber or Lyft, need to frequently solve network-matching problems, which exhibit a monotonic relationship between the state of the system (e.g., the supply of drivers and requests for rides) and the value function (e.g., platform profit), as well as the policy (e.g., pricing schedule). For instance, increasing the demand (i.e., the stochastic number of ride requests) would tend to drive up the optimal price, whereas an increase in supply (drivers) would reduce the price, all else equal. Furthermore, an increase in supply and/or demand should increase the firm's profit (the matching problem becomes less constrained), barring unexpected (and unusual) effects on demand or supply elasticity.

(iii) Administering a drug may be a salient strategy in an attempt to control the spread of a treatable infectious disease. This in turn may fuel drug resistance, however, because the drug could kill the susceptible strains allowing the strains which have developed resistance to become dominant strains. The problem of how to reduce the social cost of a disease which may include the desire to prolong the useful life of an effective drug (e.g., an antibiotic) is a control problem whose value function and optimal policy would usually exhibit monotonicity in its state variables and key parameters of the problem (the prevalence of the disease and effectiveness of the drug). For example, the higher the disease prevalence, the larger tends to be the social cost of the disease. Similarly, the higher the drug effectiveness, the lower is the minimum cost of the disease.

Algorithm 1: Domain-Knowledge Enhanced Policy Gradient.

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

Initialize the critic network $\hat{q}_{\pi_\theta}$ and the actor network π_θ using randomly generated parameters $\mathbf{w}$ and $\boldsymbol{\theta}$
Initialize target network $\hat{q}'_{\pi_\theta}$ and π'_θ 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., $w_k \geq w_{\min}$) **do**
    Select action $a_k = \pi_\theta(s_k) + \zeta$ 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 buffer D
    Sample a random minibatch of transitions $B = \{(s_l, a_l, r_l, s_{l+1})\}_{l=1}^{L}$ according to D
    Set
    $$y_l = \begin{cases} r_l, & \text{for terminal } s_{l+1}, \\ r_l + \gamma \hat{q}'_{\pi_\theta}(s_{l+1}, \pi'_\theta(s_{l+1}); \mathbf{w}'), & \text{for non-terminal } s_{l+1}. \end{cases}$$
    Update the critic weights
    $\mathbf{w} \in \arg\min_{\hat{\mathbf{w}}} \frac{1}{\|B\|} \sum_{l=1}^{L} \left[\left(y_l - \hat{q}_{\pi_\theta}(s_l, a_l; \hat{\mathbf{w}}) \right)^2 \right]$
    Compute the constraint-violation penalty $H(\pi_\theta(s_l))$
    Update the actor policy using sampled policy gradient:
    $\nabla_\theta \hat{J}(\pi_\theta; \mathbf{w}) =$
    $\frac{1}{\|B\|} \sum_{l=1}^{L} \left[\nabla_a \hat{q}_{\pi_\theta}(s_l, a; \mathbf{w})_{|a=\pi_\theta(s_l)} \nabla_\theta \pi_\theta(s_l) - \nabla_\theta H(\pi_\theta(s_l)) \right]$
    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

4. Results

Our numerical study contains 50 independent runs of the DDPG (non-penalized) and DKEPG (penalized) algorithms for the debt-recovery problem, each executed over 10,000 episodes. An episode consists of a full collection trajectory from the given 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 from the compact state space $\mathcal{S}$.[4] Importantly, in order not to 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 $H(\cdot)$ on learning, every pair of DDPG and DKEPG runs is seeded with an identical randomization seed and initialized using the same network weights. In our numerical experiment, we consider debt holders with similar characteristics, i.e., with fixed repayment-process parameters; for an empirical identification of an impulse-controlled Hawkes process in the debt-recovery context, see Chehrazi and Weber (2015) or Mark and Weber (2020). However, we differentiate individual accounts according to their starting position in the state space, $(\lambda_0, w_0) \in \mathbb{R}^2_+$. That is, an account perceived as being of a higher quality will have a higher starting intensity λ_0. For evaluation of the learning progress, we consider systematic learning measures linked to our objectives — policy quality, speed of convergence, and value-function interpretability. In addition, to make the result robust with respect to the entire state space as well as to demonstrate their practical applicability, we consider given metrics on a portfolio of 200 accounts (see Fig. 4a).

Fig. 5a displays the collection-performance evolution of both agents as measured in relative collected amount, averaged over all 50 runs (analogous to learning curves). Since their collection returns are 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 has been activated.

In Section 3, we provided a link between interpretability and policy monotonicity in the state-space. Fig. 3 demonstrates the intuitive meaning of interpretability. The shaded regions represent the action region $\mathcal{C}$ 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 $\mathcal{F}$, i.e., the interface between $\mathcal{C}$ and inaction region $\mathcal{I}$. 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 a systemically inconsistent policy any accounts in states s outside the closure of $\mathcal{C}$, but still in the (closed) convex hull of $\mathcal{C}$, would be discriminated against in treatments. Furthermore, given the required policy monotonicity in Eq. (12), 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. 3a.

To quantify the level of policy inconsistency described above, we develop two distinct metrics. First, we define an *interpretability index* (with respect to the policy monotonicity required in the application) as

$$I(\pi_\theta) = \frac{1}{\|\mathcal{C}\|} \int_{\mathcal{C}} \mathbb{1}_{\left\{ \left(\partial_\lambda \pi_\theta(\lambda, w) \leq \delta \right) \wedge \left(-\delta \leq \partial_w \pi_\theta(\lambda, w) \right) \right\}} ds, \tag{16}$$

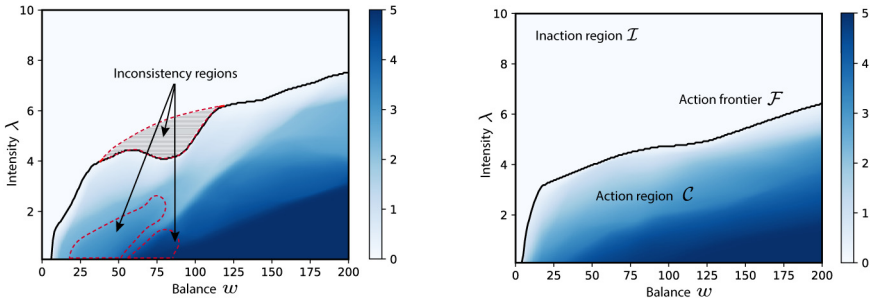
where $\delta > 0$ denotes some tolerance for non-monotonicity (zero being the most strict), and $ds = d\lambda \times dw$ denotes the standard (Lebesgue-)measure on $\mathcal{S}$. The monotonicity measure can be interpreted as a relative number of non-violations in the action set $\mathcal{C}$, i.e., how many percent of the action set is interpretable. Fig. 5b depicts the time evolution of the non-violation (compliance) metric against the 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 monotonicity) so as to connect interpretability to the agent's learning stability. 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., $\|\nabla_\theta \hat{J}(\pi_\theta^k)\| \leq \alpha$. This 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, C_ℓ is determined as

$$C_\ell = \frac{\#\text{ of policies satisfying } \left\{ I(\pi_\theta^{K^\alpha}) \geq \ell \right\}}{\#\text{ of independent training instances}}, \tag{17}$$

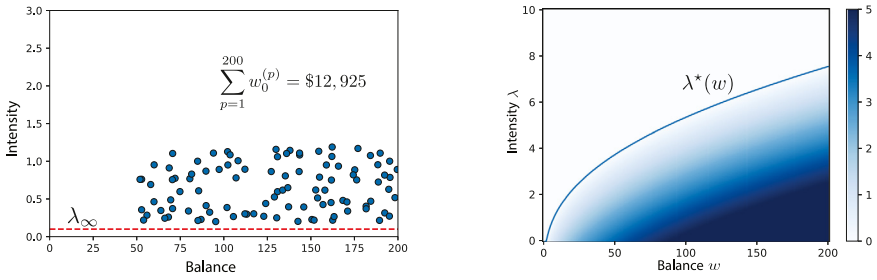
where $\ell \in [0,1]$ is a given level of interpretability. That is, we measure both the stopping time and the interpretability index at the stopping episode $k = K^\alpha$. Fig. 6a depicts this relationship on a comparison graph in the spirit of the well-known "q–q plot". We observe that both agents 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 level of interpretability $\ell = 95\%$

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



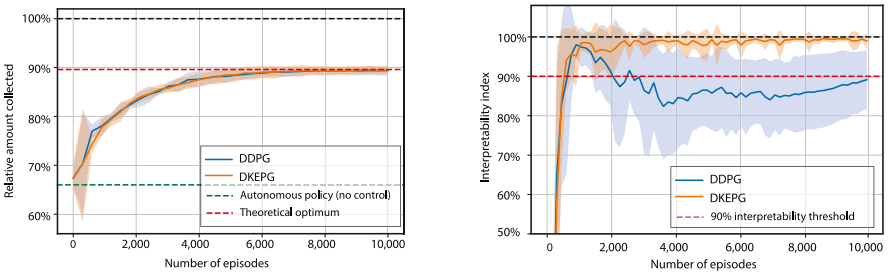
(a) Monotonicity violation (b) Fully interpretable policy

Fig. 3. Interpretability of a state-control feedback policy.



(a) Sample portfolio (b) Theoretically optimal policy

Fig. 4. (a) Portfolio of accounts used for evaluation of learning metrics. Accounts are drawn from a uniform distribution with a support on $[0.2, 1.2] \times [50, 200]$. (b) Optimal policy $\pi_\theta(\lambda, w) \in [0, 5]$.



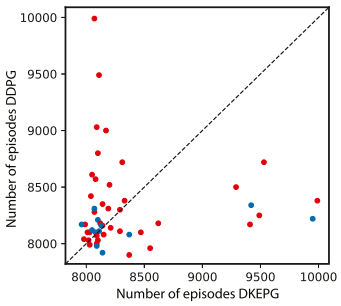
(a) Net collections of the sample portfolio (b) Interpretability index

Fig. 5. Comparison of learned policies under DKEPG and standard DDPG. The shaded area represents one standard deviation computed from 50 independent learning instances.

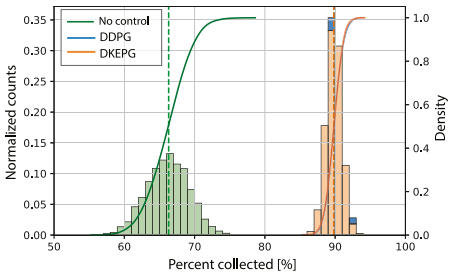
(corresponding to $I(\pi_\theta) = 95\%$), so $C_{95\%} = 11/50 = 22\%$. For $\ell = 99\%$, the systemic consistency of the non-penalized agent drops to zero. By contrast, the penalized agent terminated with an interpretable policy at the 99% level in all 50 runs, thus attaining perfect systemic consistency 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 the 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 debt-recovery problem with a value function in semi-closed form (see Fig. 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). 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. 7a we measure mean squared error (MSE) of both agent-learned frontiers $\hat{\lambda}^{\text{DDPG}}(w)$ and $\hat{\lambda}^{\text{DKEPG}}(w)$ using our 50 independent runs. Additionally, in Fig. 7b we compute the variances of $\hat{\lambda}^{\text{DDPG}}(w)$ and $\hat{\lambda}^{\text{DKEPG}}(w)$, respectively. From Fig. 7a, 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. 7b 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.

Finally, Fig. 6b showcases effects of the treatments on the revenue distribution. Indeed, the repeated reassessment of the collection strategy brought in up to 23% of extra revenue over the autonomous laissez-faire policy. Furthermore, in terms of net present value, a collection schedule following a DKEPG or DDPG produces a first-order stochastically dominant shift in the revenue distribution. In addition,

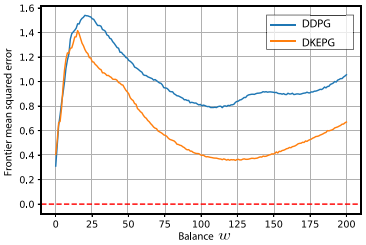


(a) Sample portfolio

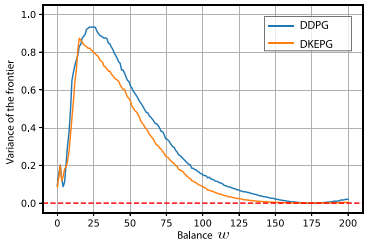


(b) Revenue binned pdf and kernel-smoothed cdf

Fig. 6. (a) Each point marks a termination of DKEPG and DDPG learning according to the stopping criterion $\|\nabla_\theta \hat{J}(\pi_\theta^k)\| \leq 10^{-3}$ colored based on the interpretability (or not) of the respective DDPG runs at termination (blue if $I(\pi_\theta) \geq 95\%$ and red otherwise). (b) Histograms represent normalized bin counts, solid lines depict the cdf and the dashed lines mark the first moment of the respective empirical distributions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) MSE of the action frontier (when fixing w) computed over 50 independent runs using the theoretically optimal action frontier (see Fig. 4)



(b) Variance of the action frontier (when fixing w) computed over 50 independent runs

Fig. 7. Comparison of the dispersion in the policy domain for DKEPG vs. DDPG.

Table 1
Average performance summary at learning termination.

	DPG	DKEPG
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

return variance decreases, thus producing a less uncertain higher-yield security, that is, a strict increase in the asset quality of an overdue account.

5. Conclusion

Contributions. This paper discusses the problem of optimal recovery of unsecured consumer debt using a novel interpretable reinforcement-learning technique, called Domain-Knowledge Enhanced (deterministic) Policy Gradient (DKEPG). This augmented reinforcement-learning approach naturally incorporates structural knowledge, thus enabling the learning of fundamentally interpretable and intelligible 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; furthermore, it provides quantifiable guarantees of interpretability in the policy space.

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 quantitative 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.

CRediT authorship contribution statement

Michael Mark: Conception and design of study, analysis and/or interpretation of data, writing – original draft, writing – review & editing. **Naveed Chehrazi:** Conception and design of study, analysis and/or interpretation of data, writing – original draft, writing – review & editing. **Huanxi Liu:** Analysis and/or interpretation of data. **Thomas A. Weber:** Conception and design of study, analysis and/or interpretation of data, writing – original draft, writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research was funded by the Swiss National Science Foundation (grant no. 105218-179175). The support is gratefully acknowledged.

Appendix. Implementation details

To ensure the reproducibility of the simulation results in Section 4 we now provide an exhaustive list of the hyperparameters used, followed by some practical considerations for the implementation of the DKEPG 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 & Weber, 2015) or the branching structure in an expectation–maximization algorithm (Mark & Weber, 2020).

A.1. Repayment-process specification

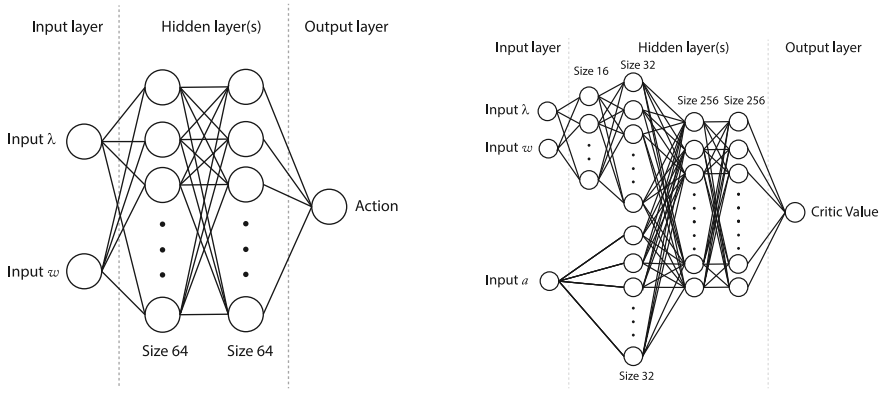
The repayment process in an MDP environment is described in Section 2.1. It features a uniform distribution $\rho_0(\lambda, w)$ of initial states on the rectangular support $S_0 = [\lambda_\infty, \lambda_{\max}] \times [w_{\min}, w_{\max}] \subset S$. The support S_0 also serves as an *invariant set* that contains all active states. That is, an action (or event) that would risk pushing the agent out of 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 debt-recovery 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 autocovariance 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 chosen as a *maximum* step size that still produces a sequence of arrivals statistically indistinguishable from 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 $\delta_{11} = 0.5$. 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 discount factor $\gamma = 0.9925$. The exact algorithm governing the MDP collections environment is sketched in Alg. 2. We note that the chosen parameters are in line with debt-recovery practice as reported by Chehrazi and Weber (2015), and the results presented in Section 4 are robust with respect to their particular values. Different runs were performed at different parameter configurations with qualitatively identical results.

A.2. Learning hyperparameters and architecture

Our actor implementation features a deep neural net (DNN) parametrization with two hidden layers, each spanning 64 individual neurons, as shown in Fig. A.8a. The critic network is also parametrized 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. A.8b. 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 update-sensitivity coefficient ξ is set to 0.005.

Algorithm 2: Discrete-Time Simulation of the Repayment Process in (1).

Algorithm parameters:
$(\lambda_0, \lambda_\infty, \kappa, \delta_1)$ — process parameters; Δt — discretization step;
π — policy
Initialize the current time $t = 0, w_k = w_0, \lambda_k = \lambda_0$
while $w_k > w_{min}$ **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
  Set $r_k = (z_k w_k - ac)$
  Set $w_k = (1 - z_k) w_k$
  Set $k = k + 1$
end



(a) Actor network (b) Critic network

Fig. A.8. DNN actor–critic architecture.

References

Chehrazi, N., Glynn, P. W., & Weber, T. A. (2019). Dynamic credit-collections optimization. *Management Science*, *65*(6), 2737–2769.

Chehrazi, N., & Weber, T. A. (2010). Monotone approximation of decision problems. *Operations Research*, *58*(4-part-2), 1158–1177.

Chehrazi, N., & Weber, T. A. (2015). Dynamic valuation of delinquent credit-card accounts. *Management Science*, *61*(12), 3077–3096.

Fedus, W., Ramachandran, P., Agarwal, R., Bengio, Y., Larochelle, H., Rowland, M., et al. (2020). Revisiting fundamentals of experience replay. In *International Conference on Machine Learning* (pp. 3061–3071).

Gupta, A., Shukla, N., Marla, L., Kolbeinsson, A., & Yellepeddi, K. (2019). How to incorporate monotonicity in deep networks while preserving flexibility? arXiv preprint arXiv:1909.10662.

Haarnoja, T., Zhou, A., Abbeel, P., & 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).

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., et al. (2017). Rainbow: Combining improvements in deep reinforcement learning. arXiv preprint arXiv:1710.02298.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Mark, M., & Weber, T. A. (2020). Robust identification of controlled Hawkes processes. *Physical Review E*, *101*(4), Article 043305.

Mitchner, M., & 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., et al. (2013). Playing Atari with deep reinforcement learning. ArXiv Preprint, arXiv:1312.5602.

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.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2021). Interpretable machine learning: Fundamental principles and 10 grand challenges. arXiv Preprint, arXiv:2103.11251.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International Conference on Machine Learning* (pp. 387–395).

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., et al. (2019). Grandmaster level in Starcraft II using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354.

You, S., Ding, D., Canini, K., Pfeifer, J., & Gupta, M. (2017). Deep lattice networks and partial monotonic functions. In *Advances in Neural Information Processing Systems* (pp. 2981–2989).