Reinforcement Learning with Non-Exponential Discounting

Matthias Schultheis
Centre for Cognitive Science
Technische Universität Darmstadt
matthias.schultheis@tu-darmstadt.de

Constantin A. Rothkopf
Centre for Cognitive Science
Technische Universität Darmstadt
constantin.rothkopf@tu-darmstadt.de

Heinz Koeppl
Centre for Cognitive Science
Technische Universität Darmstadt
heinz.koeppl@tu-darmstadt.de

Abstract

Commonly in reinforcement learning (RL), rewards are discounted over time using an exponential function to model time preference, thereby bounding the expected long-term reward. In contrast, in economics and psychology, it has been shown that humans often adopt a hyperbolic discounting scheme, which is optimal when a specific task termination time distribution is assumed. In this work, we propose a theory for continuous-time model-based reinforcement learning generalized to arbitrary discount functions. This formulation covers the case in which there is a non-exponential random termination time. We derive a Hamilton–Jacobi–Bellman (HJB) equation characterizing the optimal policy and describe how it can be solved using a collocation method, which uses deep learning for function approximation. Further, we show how the inverse RL problem can be approached, in which one tries to recover properties of the discount function given decision data. We validate the applicability of our proposed approach on two simulated problems. Our approach opens the way for the analysis of human discounting in sequential decision-making tasks.

1 Introduction

An often observed phenomenon in humans and animals is that they prefer rewards rather sooner than later [1]. It comes with no surprise that an animal searching for food aims to find it as soon as possible and employees working in a company prefer to be paid after each month instead of after each year. In behavioral experiments, it was observed that people are even willing to pay a price to receive rewards earlier [2]. It can be concluded that rewards become of less value in the future, a phenomenon which is known as discounting [2].

Modeling and experimentally inferring discounting functions, which describe how values are discounted over time, already have a long history in economics and psychology, where many different functional forms have been proposed [3, 4, 5]. While from an economic perspective, a fixed interest rate seems reasonable, leading to an exponential discount function, human behavior is oftentimes better described by a hyperbolic curve [6]. The reason for this is that human decisions are typically not consistent regarding shifting rewards in time, a phenomenon named preference reversal: We might find that a subject prefers a smaller reward on the same day over a larger reward one day after. However, when given the choice between the same smaller reward in 365 days, compared to the same

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

larger reward in 366 days, the subject is more likely to be willing to wait one day more for the larger reward.

While some literature has branded the observed discounting behavior as not being rational [7], there has been an increasing amount of work identifying circumstances under which this behavior is indeed optimal [8, 9, 10]. One widely-adopted theory rationalizing hyperbolic discounting is to assume a constant risk for the reward to become unavailable but with the risk being uncertain [11]. Under this condition, one should adapt the preference over time, as with time the expected risk decreases.

Discounting is also widely applied in the field of reinforcement learning (RL) and optimal control [12, 13, 14]. First, when modeling infinite horizon time objectives, a discount function is needed to make the expected long-term reward objective well-defined, as otherwise, it would become infinite. Second, for autonomous agents, it also makes sense to model a preference for earlier rewards in order to find shortest paths to save time and energy. Third, the discount function can be interpreted as the probability of termination inducing a specific end-time distribution [15, 12].

Despite these obvious connections, discounting models of psychology and reinforcement learning have remained mostly independent with few exceptions [16, 17]. Generalizing reinforcement learning to a broader range of discount functions would enable solutions for applications in which the end time follows a specific distribution. On the other hand, methods for determining optimal decisions in sequential decision-making tasks with general end-time distributions would provide tools which can help to explain human decision-making under uncertainty.

In this work, we present a theory for model-based reinforcement learning in continuous time based on non-exponential discount functions. First, we investigate the conditions under which the objective of maximizing the long-term reward formulated with hyperbolic discounting is well-defined. Second, we derive a Hamilton–Jacobi–Bellman (HJB)-type equation for a general discount function and describe how to solve it to obtain the optimal policy. Third, we provide an approach to tackle the inverse reinforcement learning (IRL) problem, to estimate parameters of the discount function given decision data. Finally, we show the applicability of our proposed method on two simulated problems.

2 Related work

Optimal control in continuous time and space has a long history with many classical works [18, 19, 20, 21]. Continuous-time reinforcement learning formulations have been developed [14] and various solution methods have been proposed [22, 23]. Solution approaches that solve the HJB equation directly include linearization techniques [24, 25], path integral formulations [26, 27, 12], and collocation-based methods [28]. In recent years, it has become increasingly popular to use neural networks for function approximation to solve the HJB equation [29, 30, 31, 32, 33].

Non-exponential discounting was considered in literature in behavioral economics, psychology, and neuroscience [4, 34, 2]. For humans and animals, preference reversal behavior has been described [35, 36, 5] and different functional forms for the discount function have been proposed [3, 37]. In most works, only non-sequential decision-making tasks have been studied, i.e., situations with a single decision for each independent trial. In this line of work, the method of limits has been used to elicit discount functions based on binary decisions [38, 39, 40] or an adjustment method has been applied [41]. Some methods have been proposed that specifically aim to efficiently estimate the discount function for binary choice responses [42, 43]. There have been a few exceptions for which decision trials were actually dependent but could still be considered independently for the analysis [44, 45]. Further, there has been research aiming to find rational explanations for the encountered discounting behavior [8, 9, 10], e.g., by assuming uncertainty about a constant hazard [11].

Decision processes with non-exponential time distributions have been studied in the field of semi-Markov Decision Processes (SMDPs) [46, 47, 48, 49], where transition times between discrete states can follow arbitrary distributions. MDPs with quasi-hyperbolic discounting, for which all future rewards are additionally discounted by a constant factor, have been addressed in [50, 51, 52]. They can be used to model preference reversal but are limited to the specific form assumed. Stochastic processes with quasi-hyperbolic discounting have been considered in financial economics for portfolio management [53, 54, 55, 56].

More closely related to our work, Fedus et al. [17] presented an approximate method for solving MDPs with hyperbolic discounting. They approached the problem by solving the corresponding

exponentially-discounted problem for many different discount factors and combining the results. Part of the approximation, however, is that the value function and policy are static, failing to model preference reversals over time. Another line of work considers MDPs with discount factors that are coupled to the value function to imitate hyperbolic discounting behavior [16, 57]. As time is only considered indirectly through the magnitude of the value function, these approaches cannot be used for finding the solution to a given specific discount function or eliciting the discount function from data.

Finally, inverse reinforcement learning approaches have been mainly used to learn reward functions given data [58, 59, 60, 61]. These methods were also applied to learn properties of human behavior [62, 63, 64, 65]. Other inverse approaches have focused on learning dynamics models [66] or learning rules [67] from sequential decision-making data.

3 Background

3.1 Survival analysis

In survival analysis [68], one is interested in the duration until events occur. In its classical form one considers a single event, for which the duration can be described as a continuous random variable T with cumulative distribution function $F(t) = P(T \leq t)$ and probability density function $f(t)$, where $t \in \mathbb{R}_0^+$ denotes the elapsed time. In survival analysis literature, $F(t)$ is known as the failure function and the survival function is defined as $S(t) = 1 - F(t) = P(T > t)$. The survival function is monotonically decreasing and has the properties $S(0) = 1$ and $\lim_{t\to\infty} S(t) = 0$. By the conditioning rule, one finds $P(T > t_1 \mid T > t_0) = S(t_1)/S(t_0)$ if $t_1 > t_0$. The hazard rate is defined as $\alpha(t) = \lim_{\Delta t\to 0} \frac{1}{\Delta t} P(t \leq T < t + \Delta t \mid t \leq T)$, yielding the relations

$$\alpha(t) = \lim_{\Delta t\to 0} \frac{1}{\Delta t} \frac{S(t) - S(t + \Delta t)}{S(t)} = -\frac{S'(t)}{S(t)} \quad \text{and} \quad S(t) = \exp\left(-\int_0^t \alpha(\tau)\,\mathrm{d}\tau\right). \tag{1}$$

For a constant hazard rate $\alpha(t) = \lambda$, one finds $S(t) = \exp(-\lambda t)$, which can be shown to be the unique memory-less survival function [68], i.e., $P(T > t + \Delta t \mid T > t) = P(T > \Delta t)$ for all $t, \Delta t \in \mathbb{R}_0^+$.

3.2 Discounting and preference reversal

We consider the setting in which a subject collects a single reward and shows a form of time preference, i.e., rewards are desired rather sooner than later. We model the value of a reward $r \in \mathbb{R}$ as a function $L : \mathbb{R} \times \mathbb{R}_0^+ \to \mathbb{R}$ with $L(r, t) = S(t) \cdot r$, where $S(t)$ is the discount function decreasing with time $t \in \mathbb{R}_0^+$. With the convention that $S(t_0) = 1$ and $\lim_{t\to\infty} S(t) = 0$, we can regard $S(t)$ as the survival function (Section 3.1), with the interpretation that the reward becomes unavailable with hazard rate $\alpha(t)$ [11].

When assuming a constant hazard rate $\alpha(t) = \lambda$, we say that the subject discounts exponentially, as $S(t) = \exp(-\lambda t)$. By the memory-less property, we have that if the subject prefers reward r_1 after t_1 over r_2 after t_2, she or he would remain consistent with the election if we presented the choice again later in time. On the other hand, if we assume a constant but unknown hazard rate λ with belief $p(\lambda) = \mathrm{Gamma}(\lambda; \alpha_0, \beta_0)$, we obtain a hyperbolic form for the expected survival function:

$$S(t; \alpha_0, \beta_0) = \int_\lambda \exp(-\lambda t)\, p(\lambda)\,\mathrm{d}\lambda = \frac{1}{(\frac{t}{\beta_0} + 1)^{\alpha_0}} \tag{2}$$

The posterior belief over λ at a later point in time can be derived using Bayes rule and is given by $p(\lambda \mid t) = \mathrm{Gamma}(\lambda; \alpha_0, \beta_0 + t)$. The expected hazard rate $\alpha(t)$ is given by the posterior mean,

$$\alpha(t) = \int_\lambda \lambda\, p(\lambda \mid t)\,\mathrm{d}\lambda = \frac{\alpha_0}{\beta_0 + t}. \tag{3}$$

For discount functions other than the exponential, such as the hyperbolic discount function in Eq. (2), the hazard rate varies over time and preferences among options may change.

3.3 Optimal control

In stochastic optimal control [69], we consider a Markovian system with continuous state $\mathbf{x}(t) \in \mathbb{R}^n$ evolving according to the stochastic differential equation (SDE) $\mathrm{d}\mathbf{X}(t) = f(\mathbf{X}(t), \mathbf{u}(t))\,\mathrm{d}t + G(\mathbf{X}(t), \mathbf{u}(t))\,\mathrm{d}\mathbf{W}(t)$ where $t \in \mathbb{R}_0^+$ denotes time, $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ the drift function, $G : \mathcal{X} \times \mathcal{U} \to \mathcal{X} \times \mathbb{R}^m$ the dispersion matrix, and $\mathbf{W}(t) \in \mathbb{R}^m$ m-dimensional Brownian motion. The goal of optimal control is to determine the control inputs $\mathbf{u}(t) \in \mathcal{U}$ given the current state $\mathbf{x}$ at time t, in order to maximize the expected long term discounted reward with reward function $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$. The solution is characterized by the optimal value function,

$$V^*(\mathbf{x}) = \max_{\mathbf{u}_{[t,\infty)}} \mathbb{E}\left[\int_t^\infty \exp(-\lambda(\tau - t))\, R(\mathbf{X}(\tau), \mathbf{u}(\tau))\,\mathrm{d}\tau \;\middle|\; \mathbf{X}(t) = \mathbf{x}\right], \tag{4}$$

where maximization is carried out over all trajectories $\mathbf{u}_{[t,\infty)} := \{\mathbf{u}(\tau)\}_{\tau \in [t,\infty]}$. The quantity $\exp(-\lambda t)$ is the discount factor, which ensures convergence of the integral and models a preference for earlier rewards. λ can be interpreted in terms of survival analysis as the hazard rate for termination (cf. Section 3.1). According to the principle of optimality, the stochastic Hamilton-Jacobi-Bellman (HJB) equation, given by

$$\lambda V^*(\mathbf{x}) = \max_{\mathbf{u} \in \mathcal{U}} \left[R(\mathbf{x}, u) + V^*_{\mathbf{x}}(\mathbf{x})^T f(\mathbf{x}, \mathbf{u}) + \frac{1}{2}\operatorname{tr}\left\{V^*_{\mathbf{xx}}(\mathbf{x}) G(\mathbf{x}, \mathbf{u}) G(\mathbf{x}, \mathbf{u})^T\right\}\right],$$

provides a condition for the optimal value function. Here, partial derivatives are denoted by the index notation, i.e., $V_{\mathbf{x}}$ denotes the partial derivative of V w.r.t. $\mathbf{x}$ and $V_{\mathbf{xx}}$ the respective Hessian. Note that the optimal value function depends on time only indirectly through the state $\mathbf{x}$ due to the Markov property and the memory-less discount function. This dependence also applies to the optimal policy $\pi^* : \mathcal{X} \to \mathcal{U}$, given by the maximizer of the right-hand side of the HJB equation, i.e.,

$$\pi(\mathbf{x}) = \arg\max_{\mathbf{u} \in \mathcal{U}} \left[R(\mathbf{x}, u) + V^*_{\mathbf{x}}(\mathbf{x})^T f(\mathbf{x}, \mathbf{u}) + \frac{1}{2} V^*_{\mathbf{xx}}(\mathbf{x}) G(\mathbf{x}, \mathbf{u}) G(\mathbf{x}, \mathbf{u})^T\right].$$

4 Reinforcement learning with general discount function

We consider a system as in stochastic optimal control (cf. Section 3.3) with continuous state space $\mathbb{R}^n$ and finite set of controls $\mathcal{U}$. Instead of an exponential discount function, we allow for general survival functions $S(t)$ based on a time-dependent hazard rate $\alpha(t)$. This setting generalizes the discounting formulation of Section 3.2 to sequential decisions. As we will see later, the resulting value function will become time-dependent in contrast to Section 3.3 and we can further allow for time-dependent dynamics and reward without increasing complexity. We therefore assume for the state evolution the SDE $\mathrm{d}\mathbf{X}(t) = f(\mathbf{X}(t), \mathbf{u}(t), t)\,\mathrm{d}t + G(\mathbf{X}(t), \mathbf{u}(t), t)\,\mathrm{d}\mathbf{W}(t)$ with $f : \mathcal{X} \times \mathcal{U} \times \mathbb{R}_0^+ \to \mathcal{X}$ and $G : \mathcal{X} \times \mathcal{U} \times \mathbb{R}_0^+ \to \mathcal{X} \times \mathbb{R}^m$. The objective function measuring the total expected discounted reward is given by

$$J\left(\mathbf{u}_{[0,\infty)}\right) = \mathbb{E}\left[\int_0^\infty S(\tau)\, R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau\right], \tag{5}$$

with time-dependent reward function $R : \mathcal{X} \times \mathcal{U} \times \mathbb{R}_0^+ \to \mathbb{R}$. Analogous to Section 3.3, we define the expected reward-to-go as the value function

$$V^*(\mathbf{x}, t) = \max_{\mathbf{u}_{[t,\infty)}} \mathbb{E}\left[\int_t^\infty \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau \;\middle|\; \mathbf{X}(t) = \mathbf{x}\right], \tag{6}$$

where $S(\tau)/S(t)$ is the probability of survival until time τ, conditioned on the fact that one already has survived until time t (cf. Section 3.1). In contrast to Eq. (4), the value function becomes time-dependent through the general survival function and also the optimal policy depends on time.

4.1 Technical requirements of model and discount function

First, the stochastic process defined in Section 4 must have a strongly unique solution, which is the case if f and G grow at most linearly in $\mathbf{X}$ and are Lipschitz continuous in the same variable [70]. Further, the optimal value function is only in rare cases smooth enough to be a solution in the

"classical" sense. Instead, one considers a viscosity solution [20], which satisfies the HJB equation in an appropriate generalized sense. A sufficient condition for the existence of such a solution is that f and G are continuous with bounded continuous first derivatives w.r.t. $\mathbf{X}$ and t, and bounded. R and S need to be continuous and grow at most polynomially in $\mathbf{X}$ and t in an absolute sense. More details on the existence and uniqueness of viscosity solutions can be found in [20]. A third requirement is that the integral in the definition of the value function Eq. (5) converges. For the case of a hyperbolic discount function as in Eq. (2), we find the following theorem:

Theorem 1. *Consider the hyperbolic discount function in Eq. (2). If the reward function $R(\mathbf{x}, \mathbf{u}, t)$ is bounded above for all $(\mathbf{x}, \mathbf{u}, t) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R}_0^+$, and $\alpha_0 > 1$, the value function defined in equation Eq. (6) is well-defined. If $R(\mathbf{x}, \mathbf{u}, t)$ is bounded below for all $(\mathbf{x}, \mathbf{u}, t) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R}_0^+$, and $\alpha_0 \leq 1$, the value function is not well-defined as it becomes infinite.*

Proof. See Appendix A.

In the examples considered later, we will assume a bounded reward function and a hyperbolic discount function with $\alpha_0 > 1$, for which Eq. (5) and Eq. (6) are well-defined.

4.2 HJB equation for a general discount function

In the following, we give a brief overview of the derivation of the HJB equation for a general discount function. A more detailed derivation is provided in Appendix B. First, we split the integral in Eq. (6) into two terms such that we obtain a recursive formulation of the value function:

$$V^*(\mathbf{x}, t) = \max_{\mathbf{u}_{[t,t+\Delta t]}} \mathbb{E}\left[\int_t^{t+\Delta t} \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau + \frac{S(t+\Delta t)}{S(t)} V^*(\mathbf{X}(t+\Delta t), t+\Delta t) \,\middle|\, \mathbf{X}(t) = \mathbf{x}\right]$$

For the second term in the expectation, we apply a Taylor expansion and Itô's formula [69] and obtain

$$\begin{aligned} V^*(\mathbf{X}(t+\Delta t), t+\Delta t) &= V^*(\mathbf{X}(t), t) + \int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\, f(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau \\ &+ \int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\mathbf{W}(\tau) + \int_t^{t+\Delta t} V^*_t(\mathbf{X}(\tau), \tau)\,\mathrm{d}\tau \\ &+ \int_t^{t+\Delta t} \frac{1}{2}\,\mathrm{tr}\,\left\{V^*_{\mathbf{xx}}(\mathbf{X}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)^T\right\}\mathrm{d}\tau + o(\Delta t). \end{aligned}$$

Plugging this result into the equation above, dividing both sides by Δt, and taking the limit $\Delta t \to 0$, as well as calculating the expectation w.r.t. $\mathbf{W}(t)$ leads to the desired HJB equation

$$\alpha(t) V^*(\mathbf{x}, t) = \max_{\mathbf{u}} \left[R(\mathbf{x}, \mathbf{u}, t) + V^*_t(\mathbf{x}, t) + V^*_{\mathbf{x}}(\mathbf{x}, t)\, f(\mathbf{x}, \mathbf{u}, t) + \frac{1}{2}\,\mathrm{tr}\,\left\{V^*_{\mathbf{xx}}(\mathbf{x}, t)\, G(\mathbf{x}, \mathbf{u}, t)\, G(\mathbf{x}, \mathbf{u}, t)^T\right\}\right], \tag{7}$$

where $\alpha(t)$ can be recognized to be the hazard rate corresponding to the survival function $S(t)$. We define the r.h.s. of the HJB equation in Eq. (7) without the maximization w.r.t. the action as

$$Q(\mathbf{x}, \mathbf{u}, t) := R(\mathbf{x}, \mathbf{u}, t) + V^*_t(\mathbf{x}, t) + V^*_{\mathbf{x}}(\mathbf{x}, t)\, f(\mathbf{x}, \mathbf{u}, t) + \frac{1}{2}\,\mathrm{tr}\,\left\{V^*_{\mathbf{xx}}(\mathbf{x}, t)\, G(\mathbf{x}, \mathbf{u}, t)\, G(\mathbf{x}, \mathbf{u}, t)^T\right\},$$

so that the optimal policy is given by $\pi^*(\mathbf{x}, t) = \arg\max_{\mathbf{u}} Q(\mathbf{x}, \mathbf{u}, t)$. Later on, we will consider hyperbolic discounting, for which the hazard rate $\alpha(t)$ is given by Eq. (3), i.e., $\alpha(t) = \alpha_0/(\beta_0 + t)$.

Algorithm 1: Computation of the optimal value function and policy for non-exp. discounting

Result: Optimal value function $V^{\psi}(\mathbf{x}, t)$, optimal policy $\pi^{\psi}(\mathbf{x}, t)$
Input: Parameters $\boldsymbol{\theta}$ of the discount function, system model, number of iterations K
Initialize parameters ψ of neural network for modeling $V^{\psi}(\mathbf{x}, t)$;
for $k \leftarrow 0$ **to** $K-1$ **do**
  Sample a batch of states and time points $(\hat{\mathbf{x}}, \hat{t})_{i=\{1,\dots,N\}}$;
  Push $(\hat{\mathbf{x}}, \hat{t})_{i=\{1,\dots,N\}}$ through the network to obtain $\hat{V}^{\psi}_{i=\{1,\dots,N\}}$;
  Use back-propagation to compute $\hat{V}^{\psi}_{\mathbf{x}i}$, $\hat{V}^{\psi}_{ti}$, $\hat{V}^{\psi}_{\mathbf{xx}i}$;
  Evaluate $E(\hat{V}^{\psi}_i, \hat{\mathbf{x}}_i, \hat{t}_i)$ in Eq. (8) and determine maximizing action $\hat{\mathbf{u}}_i$ for $i = 1, \dots, N$;
  Use back-propagation to compute the gradient of $\sum_i E(\hat{V}^{\psi}_i, \hat{\mathbf{x}}_i, \hat{t}_i)^2$ w.r.t. ψ ;
  Update ψ using the gradient ;
end
return $V^{\psi}(\mathbf{x}, t)$, $\pi^{\psi}(\mathbf{x}, t) = \arg\max_{\mathbf{u}} Q(\mathbf{x}, \mathbf{u}, t)$

4.3 Solving the HJB equation

In order to solve the HJB equation in Eq. (7), which is a PDE, we apply a collocation-based method [28, 29, 31]. To do so, we first reformulate the HJB equation as

$$\begin{aligned} E(V, \mathbf{x}, t) := -\alpha(t) V(\mathbf{x}, t) + \max_{\mathbf{u}} \Big[& R(\mathbf{x}, \mathbf{u}, t) + V_t(\mathbf{x}, t) + V_{\mathbf{x}}(\mathbf{x}, t)\, f(\mathbf{x}, \mathbf{u}, t) \\ & + \frac{1}{2} \operatorname{tr} \left\{ V_{\mathbf{xx}}(\mathbf{x}, t)\, G(\mathbf{x}, \mathbf{u}, t)\, G(\mathbf{x}, \mathbf{u}, t)^T \right\} \Big] = 0, \end{aligned} \tag{8}$$

and use a function approximator $V^{\psi}(\mathbf{x}, t)$ for $V^*(\mathbf{x}, t)$. The parameters ψ of the approximator can be determined by sampling random states $\hat{\mathbf{x}}_i$ and time points $\hat{t}_i$ and minimizing $\sum_i E(V^{\psi}, \hat{\mathbf{x}}_i, \hat{t}_i)^2$ w.r.t. ψ. Calculating the derivatives $V^{\psi}_{\mathbf{x}}(\mathbf{x}, t)$, $V^{\psi}_t(\mathbf{x}, t)$, $V^{\psi}_{\mathbf{xx}}(\mathbf{x}, t)$ and differentiating the objective function w.r.t. ψ is straightforward via automatic differentiation when choosing a neural network as function approximator [71]. As t is not bounded, we need to choose a reparametrization of t which maps all t to the interval $[0, 1)$ before feeding the values into the network. More details about the implemented parametrization and application of the collocation method are provided in Appendix D. The complete algorithm to learn the value function and policy is provided in Algorithm 1.

4.4 Inverse reinforcement learning for inferring the discount function

When analyzing human behavior, one might be interested in learning the underlying discount function that led to a set of observed choices. In contrast to standard inverse reinforcement learning settings, where the goal is to learn the underlying reward function R, here we assume that the reward function is given and the discount function $S(t)$ is unknown and needs to be inferred. To learn the function, we assume a parametric form with parameters $\boldsymbol{\theta}$.

As given data, we assume the states and time points at which the subject switches from one action to another one, i.e., $\mathcal{D} = \{\mathbf{x}, \mathbf{u}^-, \mathbf{u}^+, t\}_{i=1\dots N}$, describing that in state $\mathbf{x}$ at time t the action u^- is switched to u^+. The observed decision maker is assumed to use the optimal policy $\pi(\mathbf{x}, t) = \arg\max_{\mathbf{u}} Q(\mathbf{x}, \mathbf{u}, t)$. Shortly before and after switching time t, we have $Q(\mathbf{x}(t-\Delta t), \mathbf{u}^-, t-\Delta t) > Q(\mathbf{x}(t-\Delta t), \mathbf{u}^+, t-\Delta t)$ and $Q(\mathbf{x}(t+\Delta t), \mathbf{u}^-, t+\Delta t) < Q(\mathbf{x}(t+\Delta t), \mathbf{u}^+, t+\Delta t)$, respectively, indicating that before t action $\mathbf{u}^-$ is preferred over $\mathbf{u}^+$ and vice versa afterwards. By letting $\Delta t \to 0$, one finds $Q(\mathbf{x}(t), \mathbf{u}^-, t) = Q(\mathbf{x}(t), \mathbf{u}^+, t)$. A sensible objective for the inverse problem is therefore to minimize $\sum_i F(\mathbf{x}_i, \mathbf{u}^-_i, \mathbf{u}^+_i, t_i)$, defined as the squared difference of both Q values, i.e.,

$$\begin{aligned} F(\mathbf{x}, \mathbf{u}^-, \mathbf{u}^+, t) &= \left[Q(\mathbf{x}, \mathbf{u}^-, t) - Q(\mathbf{x}, \mathbf{u}^+, t) \right]^2 \\ &= \Big[R(\mathbf{x}, \mathbf{u}^+, t) - R(\mathbf{x}, \mathbf{u}^-, t) + V_{\mathbf{x}}(\mathbf{x}, t) \left(f(\mathbf{x}, \mathbf{u}^+, t) - f(\mathbf{x}, \mathbf{u}^-, t) \right) \\ &\quad + \frac{1}{2} \operatorname{tr} \left\{ V_{\mathbf{xx}}(\mathbf{x}, t) \left(G(\mathbf{x}, \mathbf{u}^+, t)\, G(\mathbf{x}, \mathbf{u}^+, t)^T - G(\mathbf{x}, \mathbf{u}^-, t)\, G(\mathbf{x}, \mathbf{u}^-, t)^T \right) \right\} \Big]^2, \end{aligned} \tag{9}$$

Algorithm 2: Computation of the gradient of F w.r.t. $\boldsymbol{\theta}$ for inferring the discount function

Result: Gradient of F (Eq. (9)) w.r.t. $\boldsymbol{\theta}$ Eq. (10)
Input: Parameters $\boldsymbol{\theta}$ of the discount function, system model, number of iterations K
Determine $V^{\psi}(\mathbf{x}, \mathbf{u}, t)$ using Algorithm 1 ;
Initialize parameters ϕ of neural network for modeling $V_{\boldsymbol{\theta}}^{\phi}(\mathbf{x}, t)$;
for $k \leftarrow 0$ **to** $K-1$ **do**
  Sample a batch of states and time points $(\hat{\mathbf{x}}, \hat{t})_{i=\{1,\dots,N\}}$;
  Push $(\hat{\mathbf{x}}, \hat{t})_{i=\{1,\dots,N\}}$ through the networks to obtain $\hat{V}_{\boldsymbol{\theta} i=\{1,\dots,N\}}^{\phi}$ and $\hat{V}_{i=\{1,\dots,N\}}^{\psi}$;
  Use back-propagation to compute $(\hat{V}_{\boldsymbol{\theta}}^{\psi})_{\mathbf{x}i}$, $(\hat{V}_{\boldsymbol{\theta}}^{\psi})_{ti}$, $(\hat{V}_{\boldsymbol{\theta}}^{\psi})_{\mathbf{xx}i}$;
  Evaluate $\sum_i H(\hat{V}_i^{\psi}, \hat{V}_{\boldsymbol{\theta} i}^{\phi}, \hat{\mathbf{x}}_i, \hat{t}_i)^2$ in Eq. (11) for $i = 1, \dots, N$;
  Use back-propagation to compute the gradient of $\sum_i H(\hat{V}_i^{\psi}, \hat{V}_{\boldsymbol{\theta} i}^{\phi}, \hat{\mathbf{x}}_i, \hat{t}_i)^2$ w.r.t. ϕ ;
  Update ϕ using the gradient ;
end
return $\mathrm{d}F/\mathrm{d}\boldsymbol{\theta}$ by evaluating Eq. (10) using $V_{\boldsymbol{\theta}}^{\phi}(\mathbf{x}, t)$

w.r.t. the parameters $\boldsymbol{\theta}$ of the discount function. The optimization needs to take Eq. (8) as a constraint into account to ensure that the HJB equation is fulfilled. Note that Eq. (9) depends indirectly on the parameters $\boldsymbol{\theta}$ through the definition of the value function. However, the objective function F could also directly depend on $\boldsymbol{\theta}$, we consider the general case in the following. In principle, the state at which the switching occurs also depends on $\boldsymbol{\theta}$. Nevertheless, we will assume that the parameters have only a minor influence on the states of switching, so that terms including the variation of $\mathbf{x}$ w.r.t. $\boldsymbol{\theta}$ can be neglected.

Gradient computation

For the minimization, it is useful to determine the gradient of the objective function F w.r.t. $\boldsymbol{\theta}$, which can be formulated using the chain rule for total derivatives as

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\theta}} &= F_{\boldsymbol{\theta}} + F_V V_{\boldsymbol{\theta}} + F_{V_{\mathbf{x}}} (V_{\mathbf{x}})_{\boldsymbol{\theta}} + F_{V_{\mathbf{xx}}} (V_{\mathbf{xx}})_{\boldsymbol{\theta}} \\ &= F_{\boldsymbol{\theta}} + F_V V_{\boldsymbol{\theta}} + F_{V_{\mathbf{x}}} (V_{\boldsymbol{\theta}})_{\mathbf{x}} + F_{V_{\mathbf{xx}}} (V_{\boldsymbol{\theta}})_{\mathbf{xx}} \,. \end{aligned} \tag{10}$$

By switching the order of the partial derivatives in the last step, we have obtained an expression for the gradient depending on $V_{\boldsymbol{\theta}}$. The partial derivative $V_{\boldsymbol{\theta}}$ measures the influence of the discounting parameters on the value function and is not straightforward to evaluate. We can get a PDE for this quantity by observing that the constraint $E(V^*, \mathbf{x}, t)$ in Eq. (8) is zero everywhere for the optimal value function V^* and therefore its derivative also needs to be zero. This method is known as the forward sensitivity method [72] and provides an additional PDE for the desired gradient $V_{\boldsymbol{\theta}}$:

$$\begin{aligned} 0 = \frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\theta}} &= E_{\boldsymbol{\theta}} + E_V V_{\boldsymbol{\theta}} + E_{V_t} (V_t)_{\boldsymbol{\theta}} + E_{V_{\mathbf{x}}} (V_{\mathbf{x}})_{\boldsymbol{\theta}} + E_{V_{\mathbf{xx}}} (V_{\mathbf{xx}})_{\boldsymbol{\theta}} \\ &= E_{\boldsymbol{\theta}} + E_V V_{\boldsymbol{\theta}} + E_{V_t} (V_{\boldsymbol{\theta}})_t + E_{V_{\mathbf{x}}} (V_{\boldsymbol{\theta}})_{\mathbf{x}} + E_{V_{\mathbf{xx}}} (V_{\boldsymbol{\theta}})_{\mathbf{xx}} =: H(V, V_{\boldsymbol{\theta}}, \mathbf{x}, t) \end{aligned} \tag{11}$$

To solve the PDE and obtain $V_{\boldsymbol{\theta}}$, we can use the same procedure as in Section 4.3. To obtain the gradient, first one has to solve the HJB equation Eq. (7) to obtain V^{ψ}, then Eq. (11) to determine $V_{\boldsymbol{\theta}}^{\phi}$. Afterwards, Eq. (10) gives the gradient to update the parameters $\boldsymbol{\theta}$ of the discount function. The quantities $V_{\boldsymbol{\theta}}$, $(V_{\boldsymbol{\theta}})_{\mathbf{x}}$, $(V_{\boldsymbol{\theta}})_{\mathbf{xx}}$ for the derivative be computed via automatic differentiation as in Section 4.3. The complete algorithm for computing the gradient is presented in Algorithm 2.

5 Experiments

We tested our derived method on two simulated problems with a random termination time following a hyperbolic survival function. First, we solved the HJB equation in Eq. (7) using the collocation method (Algorithm 1) and computed the optimal policy. Then, for applying the inverse reinforcement learning method, we evaluated the objective function in Eq. (9) and computed gradients for different parameter sets $\boldsymbol{\theta} = [\alpha_0, \beta_0]$ of the hyperbolic discount function in Eq. (2) using Algorithm 2. For

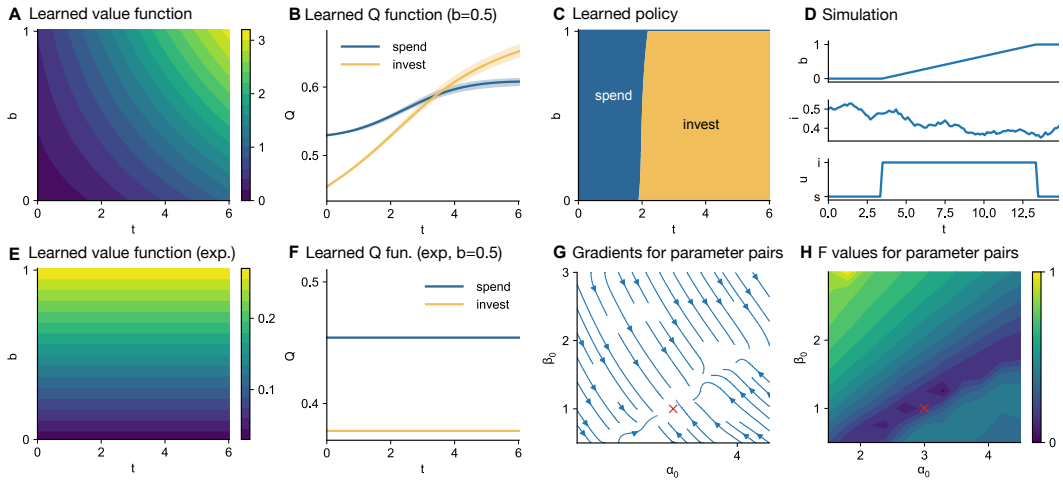


Figure 1: **Results for the investment problem. A** Learned value function for account balances and time points for hyperbolic discounting. **B** Learned Q function for both actions over time (median and quantiles for 10 runs). **C** Learned policy showing preference reversals for all account balances except for $b = 1$. **D** Simulation with account balance (top), interest rate (middle), and action (bottom) over time. **E** Learned value function for an exponential discount function, which is constant over time. **G** Learned Q function for an exponential discount function. **G** Gradients obtained for parameter pairs $[\alpha_0, \beta_0]$ of the discount function given simulated data on a 11×11 grid. The parameters used to generate the simulated trajectories are indicated by a red cross. **H** Values of the objective function F (Eq. (9)) for parameter pairs on a 11×11 grid when applying the IRL method.

generating data, we randomly sampled starting states and determined the time points at which subjects would switch their action. Afterward, the determined time points were distorted by Gaussian noise.

All proposed methods were implemented in Python using the PyTorch framework [71] and are available online[1]. The used hyperparameters are listed in Appendix F. As tasks, we considered an investment problem and a problem controlling a point on a line. In the following we provide a brief overview of the considered problems, more details can be found in Appendix E.

5.1 Experimental tasks

In the investment problem, we model a subject having to decide whether to invest her or his income in the bank account leading to future interests (rewards) or to spend the money for immediate reward. We model the state as the current balance of the bank account as well as the current interest rate. When the money is spent (action *spend*), the subject receives rewards with a rate of 0.1 but the balance of the account remains unchanged. When the income is invested (action *invest*), the balance of the bank account increases with a rate of 0.1 but there is no additional reward. In both cases the subject receives rewards through interests, being proportional to the current balance on the bank account. We assume that the interest rate varies over time following a Gaussian diffusion model. To keep the state bounded, we model a maximum balance for the account.

In the line problem, the task is to control a point along a line. The state consists of the current position of the point. Possible actions are *left* and *right*, which move the point to the respective direction, as well as *stay*, which does not move the point at all. When moving the point, there is a Gaussian diffusion on the position and one has to pay a small action cost (negative reward). We model a state-dependent reward modeling a high reward for distant states on one side and a low reward for close states on the other side. This task can be seen as a continuous extension of the classical discounting tasks. A more detailed description of the functional form is provided in Appendix E.

[1] https://git.rwth-aachen.de/bcs/nonexp-rl

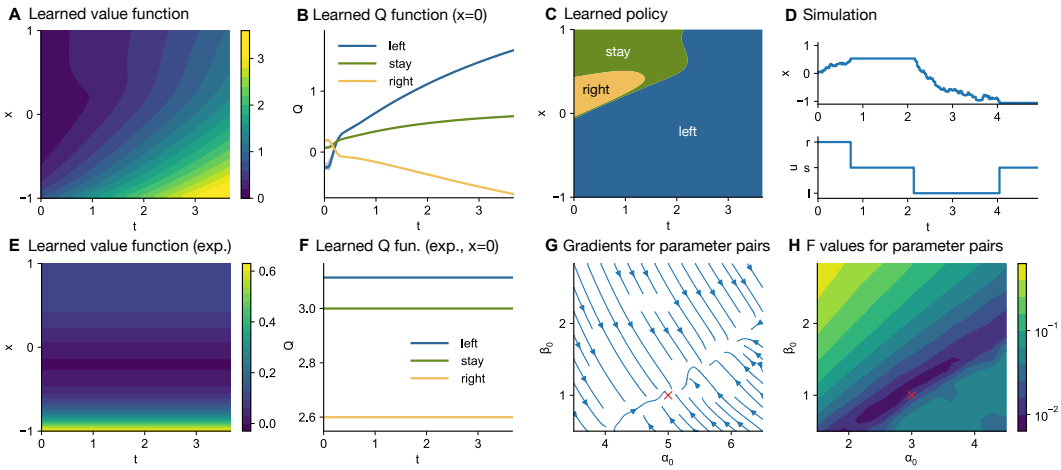


Figure 2: **Results for the line problem.** **A** Learned value function for hyperbolic discount function. **B** Learned Q function for each action over time (median and quantiles for 10 runs) **C** Learned policy showing preference reversals. **D** Simulation showing the state (top) and action (bottom) over time. **E** Learned value function for an exponential discount function. **G** Learned Q function for an exponential discount function. **G** Gradients obtained for parameter pairs $[\alpha_0, \beta_0]$ on a 15×15 grid given simulated data. The parameter used to generate the simulated trajectories is indicated by a red cross. **H** Values of the objective function of the IRL methods for parameter pairs on a 15×15 grid.

5.2 Results

The proposed method in Algorithm 1 produces plausible value functions for the considered problems. Figure 1 A shows the learned value function for the investment problem. The values are increasing with account balance (b), as expected rewards through interests become higher. Further in time, the value also increases, as the hazard rate can be assumed to be lower and one is expected to collect rewards for a longer time. Figure 1 B and C shows the learned Q function and policy, respectively. While it is advantageous to spend the income in the beginning, preference reversal occurs when the risk of termination is assumed to be relatively low, and investing becomes more attractive. The simulation in Fig. 1 D also reflects this behavior. For comparison, when assuming exponential discounting (Fig. 1 E and F), the value and Q function do not show preference reversal and remain constant over time. With regard to the inverse approach, Fig. 1 G shows the computed gradients of the IRL objective function F in Eq. (9) for different parameter pairs. One can observe that the gradients mostly point in the direction of the parameters used for simulation, indicated by a red cross. Figure 1 H shows the evaluated IRL objective function F for the parameter pairs, correctly displaying low values in the area close to the true parameters.

For the line problem, the computed value function is depicted in Fig. 2 A. Also in this task, the learned value increases over time and is high in areas close to highly rewarding states. Figure 2 B and C shows the learned Q function and policy respectively. In the beginning when the risk could be potentially high, if not close enough to the large reward, it is best to move right to collect the smaller sooner reward. With time when the risk of termination decreases, it becomes increasingly advantageous to move left to collect the larger later reward. In between, there is a time span, during which it is best to stay: The moving cost is too high for moving right and shortly turning afterward, but the risk is not low enough for being worth it to move left to the higher reward. The simulation in Fig. 2 D presents a sampled trajectory, showing that the subject first moves right until the maximum smaller sooner reward at $x = 0.5$ is reached and stays there for some time. After the risk has decreased over time, the simulated subject finally moves right to collect the larger later reward – her or his preferences have reversed. As for the investment problem, the value and Q function (Fig. 2 E and F) do not model any time-varying behavior. With regard to the IRL problem, the gradients and function values of F of Eq. (9) in Fig. 2 G and H also lead closely to parameter values used for the simulations.

6 Conclusion

In this work, we have proposed a method for reinforcement learning with non-exponential discount functions. The approach can be used to solve decision-making problems with an arbitrary end-time distribution and to model human discounting behavior. First, we have shown the conditions for which the problem is well-defined when using a hyperbolic discount function. Then, we derived a HJB-type equation providing conditions for the optimal time-dependent value function. We presented how the obtained PDE can be approximately solved using a collocation method, leading to the optimal policy. Further, we introduced an approach for the inverse problem, in which the discount function needs to be inferred given behavioral data. The application of our methods on two simulated problems led to plausible solutions, opening the way for further applications such as the use in human experiments.

Limitations and future work In our proposed methods, we assume a finite action space. While this assumption applies to many behavioral experiments, it can be a limitation for the application to classical optimal control. To extend the method to continuous control, one has to determine how the maximization problem in the HJB equation is solved. For strictly convex action costs, there have been approaches proposed that efficiently solve the maximization in the HJB equation [31], under some conditions even in closed-form [29]. Further, for Lipschitz-continuous controls, it is possible to approximate the control dynamics to avoid solving the optimization problem [73]. Another requirement of our method is that the model needs to be known. Model-free approaches, such as TD-error learning for general discount functions could be explored, in line with work such as [16]. Further, when approaching problems with large state spaces, the proposed collocation method is likely to converge slowly. For these cases, adapted collocation methods [30] or advantage updating approaches [23] could be considered instead. To apply our method to more advanced problems in financial engineering, such as modeling stocks with discontinuous returns [74], one has to consider general jump-diffusion processes. While an extension to these models is straightforward from a theoretical point, we here left it out to keep the focus on the handling of the discount function. Regarding our inverse reinforcement learning approach, we have assumed the states and time points of the action switches to be given. While this assumption is reasonable for many human behavior experiments, it might be interesting to learn discount functions for given discretized trajectories instead. Also, incorporating an extended timing model of human decisions [75] instead of fixed-variance Gaussian diffusion would be an interesting extension.

In the future, we are planning to apply our proposed methods in human experiments to get new insights to human discounting behavior. Characterizing individual human subjects by analyzing their behavior comes with the risk to be used with negative social impact. This matter can be counteracted by collecting only anonymized data for the application of our method.

Acknowledgments and Disclosure of Funding

We thank Bastian Alt for insightful discussions and for providing feedback on the first draft. This work was supported by the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt and the project "Whitebox" funded by the Priority Program LOEWE of the Hessian Ministry of Higher Education, Science, Research and Art.

References

[1] Ted O'Donoghue and Matthew Rabin. The economics of immediate gratification. *Journal of Behavioral Decision Making*, 13(2):233–250, 2000.

[2] Shane Frederick, George Loewenstein, and Ted O'donoghue. Time discounting and time preference: A critical review. *Journal of Economic Literature*, 40(2):351–401, 2002.

[3] Todd L McKerchar, Leonard Green, Joel Myerson, T Stephen Pickford, Jade C Hill, and Steven C Stout. A comparison of four models of delay discounting in humans. *Behavioural Processes*, 81(2):256–259, 2009.

[4] Robert Henry Strotz. Myopia and inconsistency in dynamic utility maximization. *The Review of Economic Studies*, 23(3):165–180, 1955.

[5] Richard Thaler. Some empirical evidence on dynamic inconsistency. *Economics Letters*, 8(3):201–207, 1981.

[6] James E Mazur. An adjusting procedure for studying delayed reinforcement. *Quantitative Analyses of Behavior*, 5:55–73, 1987.

[7] Irving Fisher. *Theory of interest: as determined by impatience to spend income and opportunity to invest it*. Augustusm Kelly Publishers, 1930.

[8] Partha Dasgupta and Eric Maskin. Uncertainty and hyperbolic discounting. *American Economic Review*, 95(4):1290–1299, 2005.

[9] Taiki Takahashi. Loss of self-control in intertemporal choice may be attributable to logarithmic time-perception. *Medical Hypotheses*, 65(4):691–693, 2005.

[10] Debajyoti Ray and Peter Bossaerts. Positive temporal dependence of the biological clock implies hyperbolic discounting. *Frontiers in Neuroscience*, 5:2, 2011.

[11] Peter D Sozou. On hyperbolic discounting and uncertain hazard rates. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1409):2015–2020, 1998.

[12] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[13] Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[14] Kenji Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

[15] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[16] William H Alexander and Joshua W Brown. Hyperbolically discounted temporal difference learning. *Neural Computation*, 22(6):1511–1527, 2010.

[17] William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

[18] Ruslan L Stratonovich. Conditional Markov processes and their application to the theory of optimal control. 1968.

[19] Harold J Kushner and Paul G Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24. Springer Science & Business Media, 2001.

[20] Wendell H Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[21] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. CRC Press, 1987.

[22] Kyriakos G Vamvoudakis and Frank L Lewis. Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

[23] Leemon C Baird. Reinforcement learning in continuous time: Advantage updating. In *IEEE International Conference on Neural Networks*, volume 4, pages 2448–2453. IEEE, 1994.

[24] David H Jacobson. New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach. *Journal of Optimization Theory and Applications*, 2(6): 411–440, 1968.

[25] Yuval Tassa, Tom Erez, and William Smart. Receding horizon differential dynamic programming. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[26] Hilbert J Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11011, 2005.

[27] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. *Journal of Machine Learning Research*, 11:3137–3181, 2010.

[28] Alex Simpkins and Emanuel Todorov. Practical numerical methods for stochastic optimal control of biological systems in continuous time and space. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 212–218. IEEE, 2009.

[29] Yuval Tassa and Tom Erez. Least squares solutions of the HJB equation with neural network value-function approximators. *IEEE Transactions on Neural Networks*, 18(4):1031–1041, 2007.

[30] Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[31] Michael Lutter, Boris Belousov, Kim Listmann, Debora Clever, and Jan Peters. HJB optimal feedback control with deep differential value functions and action constraints. In *Conference on Robot Learning*, pages 640–650. PMLR, 2020.

[32] Jiequn Han, Arnulf Jentzen, and E Weinan. Solving high-dimensional partial differential equations using deep learning. volume 115, pages 8505–8510. National Academy of Sciences, 2018.

[33] Bastian Alt, Matthias Schultheis, and Heinz Koeppl. POMDPs in continuous time and discrete spaces. In *Advances in Neural Information Processing Systems*, volume 33, pages 13151–13162. Curran Associates, Inc., 2020.

[34] Richard H Thaler and Hersh M Shefrin. An economic theory of self-control. *Journal of Political Economy*, 89(2):392–406, 1981.

[35] George Ainslie and Richard J Herrnstein. Preference reversal and delayed reinforcement. *Animal Learning & Behavior*, 9(4):476–482, 1981.

[36] Leonard Green, Nathanael Fristoe, and Joel Myerson. Temporal discounting and preference reversals in choice between delayed outcomes. *Psychonomic Bulletin & Review*, 1(3):383–389, 1994.

[37] Steffen Andersen, Glenn W Harrison, Morten I Lau, and E Elisabet Rutström. Discounting behavior: A reconsideration. *European Economic Review*, 71:15–33, 2014.

[38] Jerry B Richards, Suzanne H Mitchell, Harriet De Wit, and Lewis S Seiden. Determination of discount functions in rats with an adjusting-amount procedure. *Journal of the Experimental Analysis of Behavior*, 67(3):353–366, 1997.

[39] Maribeth Coller and Melonie B Williams. Eliciting individual discount rates. *Experimental Economics*, 2 (2):107–127, 1999.

[40] Steffen Andersen, Glenn W Harrison, Morten I Lau, and E Elisabet Rutström. Eliciting risk and time preferences. *Econometrica*, 76(3):583–618, 2008.

[41] Jawwad Noor. Hyperbolic discounting and the standard model: eliciting discount functions. *Journal of Economic Theory*, 144(5):2077–2083, 2009.

[42] Daniel R Cavagnaro, Gabriel J Aranovich, Samuel M McClure, Mark A Pitt, and Jay I Myung. On the functional form of temporal discounting: an optimized adaptive test. *Journal of Risk and Uncertainty*, 52 (3):233–254, 2016.

[43] Jorge Chang, Jiseob Kim, Byoung-Tak Zhang, Mark A Pitt, and Jay I Myung. Modeling delay discounting using gaussian process with active learning. In *CogSci*, pages 1479–1485, 2019.

[44] Nicolas Schweighofer, Kazuhiro Shishida, Cheol E Han, Yasumasa Okamoto, Saori C Tanaka, Shigeto Yamawaki, and Kenji Doya. Humans can adopt optimal discounting strategy under real-time constraints. *PLoS Computational Biology*, 2(11):e152, 2006.

[45] Maayke Suzanne Seinstra, Manuela Sellitto, and Tobias Kalenscher. Rate maximization and hyperbolic discounting in human experiential intertemporal decision making. *Behavioral Ecology*, 29(1):193–203, 2018.

[46] Ronald A Howard. Semi-Markovian decision processes. *Bulletin of the International Statistical Institute*, 40(2):625–652, 1963.

[47] V S Korolyuk, S M Brodi, and A F Turbin. Semi-Markov processes and their applications. *Journal of Soviet Mathematics*, 4(3):244–280, 1975.

[48] Sheldon M Ross. Average cost semi-Markov decision processes. *Journal of Applied Probability*, 7(3): 649–656, 1970.

[49] Steven Bradtke and Michael Duff. Reinforcement learning methods for continuous-time Markov decision problems. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994.

[50] Anna Jaśkiewicz and Andrzej S Nowak. Markov decision processes with quasi-hyperbolic discounting. *Finance and Stochastics*, 25(2):189–229, 2021.

[51] Tomas Bjork and Agatha Murgoci. A general theory of Markovian time inconsistent stochastic control problems. *SSRN*, 2010.

[52] Tomas Björk and Agatha Murgoci. A theory of Markovian time-inconsistent stochastic control in discrete time. *Finance and Stochastics*, 18(3):545–592, 2014.

[53] Steven R Grenadier and Neng Wang. Investment under uncertainty and time-inconsistent preferences. *Journal of Financial Economics*, 84(1):2–39, 2007.

[54] Ziran Zou, Shou Chen, and Lei Wedge. Finite horizon consumption and portfolio decisions with stochastic hyperbolic discounting. *Journal of Mathematical Economics*, 52:70–80, 2014.

[55] A Chunxiang, Zhongfei Li, and Fan Wang. Optimal investment strategy under time-inconsistent preferences and high-water mark contract. *Operations Research Letters*, 44(2):212–218, 2016.

[56] Shumin Chen, Zhongfei Li, and Yan Zeng. Optimal dividend strategy for a general diffusion process with time-inconsistent preferences and ruin penalty. *Journal on Financial Mathematics*, 9(1):274–314, 2018.

[57] Hado van Hasselt, John Quan, Matteo Hessel, Zhongwen Xu, Diana Borsa, and André Barreto. General non-linear Bellman equations. *arXiv preprint arXiv:1907.03687*, 2019.

[58] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, volume 1, 2000.

[59] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, volume 5, 2004.

[60] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 23, pages 1433–1438, 2008.

[61] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-Learn: Inverse soft-q learning for imitation. In *Advances in Neural Information Processing Systems*, volume 34, pages 4028–4039. Curran Associates, Inc., 2021.

[62] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion – an inverse optimal control approach. *Autonomous Robots*, 28(3):369–383, 2010.

[63] Constantin A Rothkopf and Dana H Ballard. Modular inverse reinforcement learning for visuomotor behavior. *Biological cybernetics*, 107(4):477–490, 2013.

[64] Katharina Muelling, Abdeslam Boularias, Betty Mohler, Bernhard Schölkopf, and Jan Peters. Learning strategies in table tennis using inverse reinforcement learning. *Biological Cybernetics*, 108(5):603–619, 2014.

[65] Matthias Schultheis, Dominik Straub, and Constantin A Rothkopf. Inverse optimal control adapted to the noise characteristics of the human sensorimotor system. In *Advances in Neural Information Processing Systems*, volume 34, pages 9429–9442. Curran Associates, Inc., 2021.

[66] Matthew Golub, Steven Chase, and Byron Yu. Learning an internal dynamics model from control demonstration. In *International Conference on Machine Learning*, pages 606–614. PMLR, 2013.

[67] Zoe Ashwood, Nicholas A. Roy, Ji Hyun Bak, and Jonathan W Pillow. Inferring learning rules from animal decision-making. In *Advances in Neural Information Processing Systems*, volume 33, pages 3442–3453. Curran Associates, Inc., 2020.

[68] Odd Aalen, Ornulf Borgan, and Hakon Gjessing. *Survival and event history analysis: a process point of view*. Springer Science & Business Media, 2008.

[69] Floyd B Hanson. *Applied stochastic processes and control for jump-diffusions: modeling, analysis and computation*. SIAM, 2007.

[70] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[71] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[72] Herschel Rabitz, Mark Kramer, and D Dacol. Sensitivity analysis in chemical kinetics. *Annual Review of Physical Chemistry*, 34(1):419–461, 1983.

[73] Jeongho Kim, Jaeuk Shin, and Insoon Yang. Hamilton-Jacobi deep Q-learning for deterministic continuous-time systems with Lipschitz continuous controls. *Journal of Machine Learning Research*, 22:206–1, 2021.

[74] Robert C Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2):125–144, 1976.

[75] Vijay Mohan K Namboodiri, Stefan Mihalas, Tanya Marton, and Marshall Gilmer Hussain Shuler. A general theory of intertemporal decision-making and the perception of time. *Frontiers in Behavioral Neuroscience*, 8:61, 2014.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] All contributions are provided in Section 4 with applications to experiments in Section 5.
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In the intro text in Section 4.
   (b) Did you include complete proofs of all theoretical results? [Yes] Complete proofs are provided in Appendix A and Appendix B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see Appendix F
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See error bars for learned Q functions in Section 5.2
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix F.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Pytorch was used and referenced.
   (b) Did you mention the license of the assets? [Yes] see Appendix F.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code is provided in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We have not used data of other people.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We have not used data of other people.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix
Reinforcement Learning
with Non-Exponential Discounting

A Convergence proof for the value function under hyperbolic discounting

In the following, we assume a hyperbolic survival function as presented in Eq. (2), i.e.,

$$S(t;\alpha,\beta) = \frac{1}{(\frac{t}{\beta}+1)^{\alpha}}.$$

Part I *If the reward function $R(\mathbf{x},\mathbf{u},t)$ is bounded above for all $(\mathbf{x},\mathbf{u},t) \in \mathcal{X}\times\mathcal{U}\times\mathbb{R}_0^+$, and $\alpha_0 > 1$, the value function defined in equation Eq. (6) is well-defined.*

We denote the supremum of the reward function $R(\mathbf{x},\mathbf{u},t)$ for all $(\mathbf{x},\mathbf{u},t) \in \mathcal{X}\times\mathcal{U}\times\mathbb{R}_0^+$ by r_{sup}. We find

$$\begin{aligned}
V^*(\mathbf{x},t) &= \max_{\mathbf{u}_{[t,\infty)}} \mathbb{E}\left[\int_t^\infty \frac{S(\tau)}{S(t)}\, R(\mathbf{X}(\tau),\mathbf{u}(\tau),\tau)\,\mathrm{d}\tau \,\middle|\, \mathbf{X}(t)=\mathbf{x}\right] \\
&\leq \int_t^\infty \frac{S(\tau)}{S(t)}\, r_{\text{sup}}\,\mathrm{d}\tau \\
&= \frac{r_{\text{sup}}}{S(t)} \int_t^\infty S(\tau)\,\mathrm{d}\tau \\
&= \frac{r_{\text{sup}}}{S(t)} \int_t^\infty \frac{1}{\left(\frac{\tau}{\beta}+1\right)^{\alpha}}\,\mathrm{d}\tau \\
&\leq \frac{r_{\text{sup}}}{S(t)} \int_t^\infty \frac{1}{\left(\frac{\tau}{\beta}\right)^{\alpha}}\,\mathrm{d}\tau \\
&= \frac{\beta^\alpha\, r_{\text{sup}}}{S(t)} \int_t^\infty \frac{1}{\tau^\alpha}\,\mathrm{d}\tau \\
&= \frac{\beta^\alpha\, r_{\text{sup}}}{S(t)} \left[\frac{\tau^{1-\alpha}}{1-\alpha}\right]_{\tau=t}^{\infty} \\
&= \frac{\beta^\alpha\, r_{\text{sup}}}{S(t)\,(1-\alpha)} \left[\tau^{1-\alpha}\right]_{\tau=t}^{\infty},
\end{aligned}$$

which is finite for $\alpha > 1$.

Part II *If $R(\mathbf{x},\mathbf{u},t)$ is bounded below for all $(\mathbf{x},\mathbf{u},t) \in \mathcal{X}\times\mathcal{U}\times\mathbb{R}_0^+$, and $\alpha_0 \leq 1$, the value function defined in equation Eq. (6) is not well-defined.*

We denote the infimum of the reward function $R(\mathbf{x}, \mathbf{u}, t)$ for all $(\mathbf{x}, \mathbf{u}, t) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R}_0^+$ by r_{inf}. We find

$$\begin{aligned}
V^*(\mathbf{x}, t) &= \max_{\mathbf{u}_{[t,\infty)}} \mathbb{E}\left[\int_t^\infty \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau \,\middle|\, \mathbf{X}(t) = \mathbf{x}\right] \\
&\geq \int_t^\infty \frac{S(\tau)}{S(t)} r_{\text{inf}}\,\mathrm{d}\tau \\
&= \frac{r_{\text{inf}}}{S(t)} \int_t^\infty S(\tau)\,\mathrm{d}\tau \\
&= \frac{r_{\text{inf}}}{S(t)} \int_t^\infty \frac{1}{\left(\frac{\tau}{\beta}+1\right)^\alpha}\,\mathrm{d}\tau \\
&= \frac{r_{\text{inf}}}{S(t)} \int_t^\infty \frac{1}{\left(\frac{\tau+\beta}{\beta}\right)^\alpha}\,\mathrm{d}\tau \\
&= \frac{\beta^\alpha\, r_{\text{inf}}}{S(t)} \int_t^\infty \frac{1}{(\tau+\beta)^\alpha}\,\mathrm{d}\tau \\
&= \frac{\beta^\alpha\, r_{\text{inf}}}{S(t)} \int_{t+\beta}^\infty \frac{1}{\tau^\alpha}\,\mathrm{d}\tau \\
&= \frac{\beta^\alpha\, r_{\text{inf}}}{S(t)} \left[\frac{1}{\tau^\alpha}\right]_{\tau=t+\beta}^\infty \\
&= \frac{\beta^\alpha\, r_{\text{inf}}}{S(t)} \left[\frac{\tau^{1-\alpha}}{1-\alpha}\right]_{\tau=t+\beta}^\infty \\
&= \frac{\beta^\alpha\, r_{\text{inf}}}{S(t)\,(1-\alpha)} \left[\tau^{1-\alpha}\right]_{\tau=t+\beta}^\infty,
\end{aligned}$$

in which the integral diverges for $\alpha \leq 1$.

B Full derivation of the HJB equation

In this section, we provide a full derivation for the HJB equation introduced in Section 4.2. We start with the value function defined in Eq. (6), i.e.,

$$V^*(\mathbf{x}, t) = \max_{\mathbf{u}_{[t,\infty)}} \mathbb{E}\left[\int_t^\infty \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau \,\middle|\, \mathbf{X}(t) = \mathbf{x}\right].$$

First, we split the integral into two terms and obtain

$$\begin{aligned}
V^*(\mathbf{x}, t) = \max_{\mathbf{u}_{[t,t+\Delta t]}} \mathbb{E}\Bigg[&\int_t^{t+\Delta t} \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau \\
&+ \int_{t+\Delta t}^\infty \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau \,\Bigg|\, \mathbf{X}(t) = \mathbf{x}\Bigg].
\end{aligned}$$

By identifying the second term as the value function of state $x(t+\Delta t)$ at time $t+\Delta t$, we obtain the recursive formulation

$$\begin{aligned}
V^*(\mathbf{x}, t) = \max_{\mathbf{u}_{[t,t+\Delta t]}} \mathbb{E}\Bigg[&\int_t^{t+\Delta t} \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau \\
&+ \frac{S(t+\Delta t)}{S(t)} V^*(\mathbf{X}(t+\Delta t), t+\Delta t) \,\Bigg|\, \mathbf{X}(t) = \mathbf{x}\Bigg].
\end{aligned}$$

Consider a small Δt, then the first term evaluates to

$$\int_t^{t+\Delta t} \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau = R(\mathbf{X}(t), \mathbf{u}(t), t) \cdot \Delta t + o(\Delta t).$$

For the second term, we apply a Taylor expansion and get

$$V^*(\mathbf{X}(t+\Delta t), t+\Delta t) = V^*(\mathbf{X}(t), t) + \int_t^{t+\Delta t} \frac{\mathrm{d}}{\mathrm{d}\tau} V^*(\mathbf{X}(\tau), \tau)\,\mathrm{d}\tau + o(\Delta t)$$
$$= V^*(\mathbf{X}(t), t) + \int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\,\mathrm{d}\mathbf{X}(\tau) + \int_t^{t+\Delta t} V^*_t(\mathbf{X}(\tau), \tau)\,\mathrm{d}\tau + o(\Delta t).$$

Here, the second term can be evaluated using Itô's formula as

$$\int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\,\mathrm{d}\mathbf{X}(\tau) = \int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\, f(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau$$
$$+ \int_t^{t+\Delta t} \frac{1}{2}\,\mathrm{tr}\,\left\{V^*_{\mathbf{xx}}(\mathbf{X}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)^T\right\}\mathrm{d}\tau$$
$$+ \int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\mathbf{W}(\tau) + o(\Delta t).$$

Plugging in these terms into the equation above and dividing both sides by Δt yields

$$\frac{1 - \frac{S(t+\Delta t)}{S(t)}}{\Delta t} V^*(\mathbf{X}(t), t) = \max_{\mathbf{u}_{[t,t+\Delta t]}} \mathbb{E}\left[\frac{1}{\Delta t}\int_t^{t+\Delta t} \frac{S(\tau)}{S(t)} R(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau\right.$$
$$+ \frac{1}{\Delta t}\int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\, f(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\tau + \int_t^{t+\Delta t} V^*_t(\mathbf{X}(\tau), \tau)\,\mathrm{d}\tau$$
$$+ \frac{1}{\Delta t}\int_t^{t+\Delta t} \frac{1}{2}\,\mathrm{tr}\,\left\{V^*_{\mathbf{xx}}(\mathbf{X}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)^T\right\}\mathrm{d}\tau$$
$$\left. + \frac{1}{\Delta t}\int_t^{t+\Delta t} V^*_{\mathbf{x}}(\mathbf{X}(\tau), \tau)\, G(\mathbf{X}(\tau), \mathbf{u}(\tau), \tau)\,\mathrm{d}\mathbf{W}(\tau) + \frac{o(\Delta t)}{\Delta t} \,\middle|\, \mathbf{X}(t) = \mathbf{x}\right].$$

The factor on the l.h.s. in the limit $\Delta t \to 0$ can be recognized to be the hazard rate (cf. Eq. (1)),

$$\lim_{\Delta t \to 0} \frac{1 - \frac{S(t+\Delta t)}{S(t)}}{\Delta t} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{S(t) - S(t+\Delta t)}{\Delta t} = \alpha(t).$$

Taking the limit $\Delta t \to 0$ on both sides and calculating the expectation w.r.t. $\mathbf{W}(t)$, we obtain the HJB equation

$$\alpha(t) V^*(\mathbf{x}, t) = \max_{\mathbf{u}} \left[R(\mathbf{x}, \mathbf{u}, t) + V^*_t(\mathbf{x}, t) + V^*_{\mathbf{x}}(\mathbf{x}, t)\, f(\mathbf{x}, \mathbf{u}, t)\right.$$
$$\left. + \frac{1}{2}\,\mathrm{tr}\,\left\{V^*_{\mathbf{xx}}(\mathbf{x}, t)\, G(\mathbf{x}, \mathbf{u}, t)\, G(\mathbf{x}, \mathbf{u}, t)^T\right\}\right].$$

C Bellman equation for discrete time

We consider the discrete-time setting, in which the objective is given as

$$J(\mathbf{u}_0, \mathbf{u}_1, \dots) = \mathbb{E}\left[\sum_{\tau=0}^{\infty} S(\tau)\, R(\mathbf{X}_\tau, \mathbf{u}_\tau, \tau)\right].$$

As in the continuous-time case, we can define the value function as

$$V(\mathbf{x}, t) = \max_{\mathbf{u}_t, \mathbf{u}_{t+1}, \dots} \mathbb{E}\left[\sum_{\tau=t}^{\infty} \frac{S(\tau)}{S(t)}\, R(\mathbf{X}_\tau, \mathbf{u}_\tau, \tau) \,\middle|\, \mathbf{X}_t = \mathbf{x}\right].$$

By identifying the recursive definition of the value function and evaluating terms, we obtain the Bellman equation

$$V(\mathbf{x}, t) = \max_{\mathbf{u}} \left\{R(\mathbf{x}, \mathbf{u}, t) + \lambda(t)\, \mathbb{E}\left[V(\mathbf{X}_{t+1}, t+1) \mid \mathbf{X}_t = \mathbf{x}\right]\right\},$$

with $\lambda(t) = S(t+1)/S(t)$ being the hazard probability at time t.

D Value function approximation and collocation method

In the collocation method in Algorithm 1, we need to sample random states $\hat{\mathbf{x}}_i$ and time points $\hat{t}_i$ for minimizing $\sum_i E(V^\psi, \hat{\mathbf{x}}_i, \hat{t}_i)^2$. If we assume a bounded state space $\mathcal{X} \in \mathbb{R}$, we can sample $\hat{\mathbf{x}}_i$ uniformly from this space. The time points $\hat{t}_i \in \mathbb{R}_0^+$ can be sampled from an exponential distribution. To do so, we first draw $\hat{y}_i \sim \mathrm{Uniform}(0,1)$ and compute $\hat{t}_i = -\log(1-\hat{y}_i)/\lambda$. To feed a normalized value of time into the network, we use $\hat{y}_i$ instead of $\hat{t}_i$ as input to the network. We denote the value function network depending on y by $\tilde{V}(\mathbf{x}, y)$. Given a specific time value t, we can compute its representation via $y(t) = 1 - \exp(-\lambda t)$.

When computing the partial derivative V_t, we have to take this reparametrization into account. By the chain rule, we find

$$V_t(\mathbf{x}, t) = \tilde{V}_y(\mathbf{x}, t)\, y_t(t),$$

for which we have with the chosen parametrization

$$y_t(t) = \lambda \exp(-\lambda t).$$

In general, there are multiple solutions to the HJB equation and the encountered solution depends on the initialization of the function approximator [29, 31]. In other work, this problem has been dealt with by omitting stochastic terms in the first episodes of training or annealing the discount factor [29, 31, 33]. We adopt the second approach and move from short to far-sighted discounting to converge to the desired solution. For hyperbolic discounting, we initially add an offset to α_0, leading to a high expected hazard rate. Over time, we decrease the offset to converge to the desired solution.

E Experiments

Investment problem

- State space $\mathcal{X} = [0,1] \times [0,1]$, modeling account balance and interest rate, i.e., $\mathbf{x} = [x_b, x_i]$
- Action space $\mathcal{U} = \{\textit{spend}, \textit{invest}\}$
- Dynamics model

$$f(\mathbf{x}, \mathbf{u}) = \begin{cases} [0,0]^T & \text{if } \mathbf{u} \text{ is } \textit{spend} \\ [0.1, 0]^T & \text{if } \mathbf{u} \text{ is } \textit{invest} \end{cases}$$

$$G(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} 0 & 0 \\ 0 & 0.01 \end{pmatrix}$$

- Reward function

$$R(\mathbf{x}, \mathbf{u}) = R^{\mathbf{x}}(\mathbf{x}) + R^{\mathbf{u}}(\mathbf{u})$$

$$R^{\mathbf{x}}([x_b, x_i]) = x_b \cdot x_i$$

$$R^{\mathbf{u}}(\mathbf{u}) = \begin{cases} 0.1 & \text{if } \mathbf{u} \text{ is } \textit{spend} \\ 0 & \text{if } \mathbf{u} \text{ is } \textit{invest} \end{cases}$$

- Initial belief of hazard rate $\alpha_0 = 3, \beta_0 = 1$ (visualized in Fig. 3)

Line problem

- State space $\mathcal{X} = [-1, 1]$
- Action space $\mathcal{U} = \{\textit{left}, \textit{stay}, \textit{right}\}$
- Dynamics model

$$f(\mathbf{x}, \mathbf{u}) = \begin{cases} -1 & \text{if } \mathbf{u} \text{ is } \textit{left} \\ 0 & \text{if } \mathbf{u} \text{ is } \textit{stay} \\ 1 & \text{if } \mathbf{u} \text{ is } \textit{right} \end{cases}$$

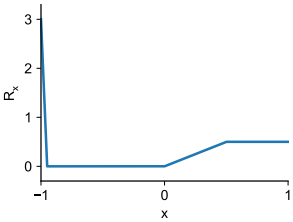
$$G(\mathbf{x}, \mathbf{u}) = \begin{cases} 0.05 & \text{if } \mathbf{u} \in \{\textit{left}, \textit{right}\} \\ 0 & \text{if } \mathbf{u} \text{ is } \textit{stay} \end{cases}$$

- Reward function

$$R(\mathbf{x}, \mathbf{u}) = R^{\mathbf{x}}(\mathbf{x}) + R^{\mathbf{u}}(\mathbf{u})$$

$$R^{\mathbf{x}}(\mathbf{x}) = \begin{cases} 0.5 & \text{if } \mathbf{x} \geq 0.5 \\ \mathbf{x} & \text{if } 0 \leq \mathbf{x} < 0.5 \\ 0 & \text{if } -0.95 \leq \mathbf{x} < 0 \\ -60\mathbf{x} - 57 & \text{if } \mathbf{x} < 0.95 \end{cases}$$

$$R^{\mathbf{u}}(\mathbf{u}) = \begin{cases} 0.1 & \text{if } \mathbf{u} \in \{\textit{left}, \textit{right}\} \\ 0 & \text{if } \mathbf{u} \text{ is } \textit{stay} \end{cases}$$



- Initial belief of hazard rate $\alpha_0 = 5, \beta_0 = 1$ (visualized in Fig. 3)

F Hyperparameters, implementation, and computing resources

Throughout the experiments, we have used the following hyperparameters:

- The neural networks are parametrized as

```
layers = (nn.Linear(input_dim, layer_size),
          nn.Sigmoid(),
          nn.Linear(layer_size, layer_size),
          nn.Sigmoid(),
          nn.Linear(layer_size, output_dim))
model = nn.Sequential(*layers)
```

- For the neural network representing V, we used

```
input_dim = x_dim + 1
output_dim = 1
```

- For the neural network representing V_θ, we used

```
input_dim = x_dim + 1
output_dim = theta_dim
```

- We set $\lambda = 0.2$.
- For the collocation method, we used 10.000 samples in each iteration and 125.000 episodes for the investment problem and 100.000 episodes for the line problem. The initial offset of α_0 was set to 50 and linearly decreased to zero over 50.000 episodes.
- We used Adam optimizer with learning rate 0.003.
- For the runs with exponential discounting, the mean of the initial belief over the hazard rate was taken for λ, i.e., 3 for the investment problem and 5 for the line problem.

More information about Implementation and computing resources:

- Methods were implemented in Python using the PyTorch framework [71], which has been published under a BSD license.
- Resources used: Intel® Xeon® Platinum 9242 Processor, using 8 cores per run.
- Network training took ~50 min. for the investment problem and ~30 min. for the line problem.

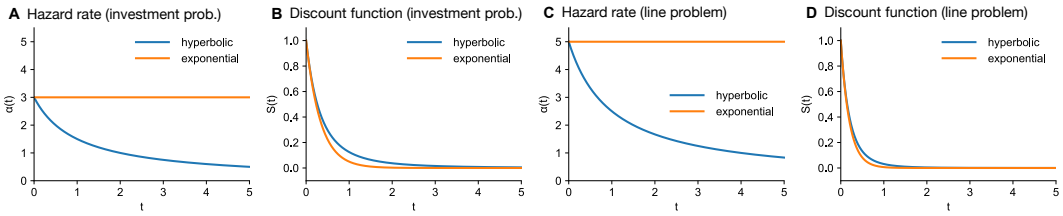


Figure 3: **Hazard rates and Discounting functions.** **A** Expected hazard rate for the investment problem. For hyperbolic discounting, the expected risk of termination is decreasing over time, while for exponential discounting, the hazard rate is constant. **B** Expected discount function for the investment problem in comparison to an exponential discount function. **C** Expected hazard rate for the line problem for hyperbolic discounting in comparison with the constant hazard rate when applying exponential discounting **D** Expected discount function for the line problem in comparison to an exponential discount function.

G Derivation of the hyperbolic discount function as uncertainty over the constant hazard rate

We assume $P(T > t \mid \lambda) = \exp(-\lambda t)$ and a belief $\lambda \sim \mathrm{Gamma}(\lambda; \alpha, \beta)$. For the expected survival function, we calculate

$$\begin{aligned} S(t) &= \int_\lambda \exp\left(-\lambda t\right) p\left(\lambda\right) \, \mathrm{d}\lambda \\ &= \int_\lambda \exp\left(-\lambda t\right) \frac{\beta^\alpha \lambda^{\alpha-1} \exp(-\beta\lambda)}{\Gamma(\alpha)} \, \mathrm{d}\lambda \\ &= \int_\lambda \frac{\beta^\alpha \lambda^{\alpha-1} \exp(-(\beta+t)\lambda)}{\Gamma(\alpha)} \, \mathrm{d}\lambda \\ &= \frac{\beta^\alpha}{(\beta+t)^\alpha} \int_\lambda \mathrm{Gamma}(\lambda; \alpha, \beta+t) \, \mathrm{d}\lambda \\ &= \frac{1}{\left(\frac{t}{\beta}+1\right)^\alpha}. \end{aligned}$$

H Interpretation of the discount factor as transition to terminal state

A Markov decision process (MDP) with discounting can be converted to an MDP without discounting by adding an additional terminal state Υ [13]. From each state with a certain probability γ, one transitions to the terminal state, and the remaining transition probabilities are renormalized. At the terminal state there is no possibility to transition to any other state and a reward of zero is given. In continuous time, the same formalization can be applied, but we consider a rate instead at which one transitions to the terminal state. Further, we assume in the following that the rate depends on time and denote it by $\lambda(t)$. The probability to be in the terminal states at time Υ is given by the cumulative distribution function (CDF),

$$P(\mathbf{X}(t) = \Upsilon) = P(T < t).$$

The probability of not having terminated yet is given by the complementary cumulative distribution function (CCDF),

$$\begin{aligned} P(\mathbf{X}(t) \neq \Upsilon) &= 1 - P(\mathbf{X}(t) = \Upsilon) \\ &= P(T \geq t) \\ &= S(t), \end{aligned}$$

which is equal to the discount function.

For a constant termination rate, one obtains the CDF and CCDF of the exponential distribution, respectively:

$$\begin{aligned} P(\mathbf{X}(t) = \Upsilon) &= \lambda \int_0^t \exp(-\lambda\tau)\,\mathrm{d}\tau \\ &= 1 - \exp(-\lambda t) \end{aligned}$$

$$\begin{aligned} P(\mathbf{X}(t) \neq \Upsilon) &= 1 - P(\mathbf{X}(t) = \Upsilon) \\ &= \exp(-\lambda t) \end{aligned}$$