Statistica Sinica Preprint No: SS-2022-0407	
Title	Reinforcement Learning via Nonparametric Smoothing in a Continuous-Time Stochastic Setting with Noisy Data
Manuscript ID	SS-2022-0407
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202022.0407
Complete List of Authors	Chenyang Jiang, Bowen Hu, Yazhen Wang and Shang Wu
Corresponding Authors	Shang Wu
E-mails	shangwu@fudan.edu.cn

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

Reinforcement Learning via Nonparametric Smoothing in a Continuous-Time Stochastic Setting with Noisy Data

Chenyang Jiang[1], Bowen Hu[1], Yazhen Wang[1], Shang Wu[2]

University of Wisconsin-Madison[1] *and Fudan University*[2]

Abstract: Reinforcement learning was developed mainly for discrete-time Markov decision processes. This paper establishes a novel learning approach based on temporal-difference and nonparametric smoothing to solve reinforcement learning problems in a continuous-time setting with noisy data, where the true model to learn is governed by an ordinary differential equation, and data samples are generated from a stochastic differential equation that is considered as a noisy version of the ordinary differential equation. Continuous-time temporal-difference learning developed for deterministic models is unstable and in fact diverges when applied to data generated from stochastic models. Furthermore, because there are measurement errors or noises in the observed data, a new reinforcement learning framework is needed to handle the learning problems with noisy data. We show that the proposed learning approach has a robust performance for learning deterministic functions based on noisy data generated from stochastic models governed by stochastic differential equations. An asymptotic theory is established for the proposed approach, and a numerical study is carried out to solve a pendulum reinforcement learning problem and check the finite sample performance of the proposed method.

Key words and phrases: HJB equation, Markov decision process, nonparametric smoothing, ordinary differential equation, policy, reinforcement learning, reward function, stochastic differential equation, value function.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

1. Introduction

As one of three basic machine learning paradigms, reinforcement learning (RL) deals with how an agent takes action in an environment to maximize rewards. It is currently very popular and has drawn significant attention. Its many applications include the famous Alpha Go (Silver et al., 2017) and Deep Q-Network (DQN) (Mnih et al., 2015). Most of RL methods apply to discrete-time Markov decision processes (MDP) (Sutton and Barto, 2018). However, many real problems, such as those involving financial assets and physical processes, are traditionally described in a continuous-time stochastic framework and naturally fit to the continuous-time RL paradigm. A large volume of literature is devoted to study these problems in stochastic control and applied mathematics, yet there is relatively little work on the study of these problems from the RL perspective (Bertsekas, 2017; Kushner and Dupuis, 2001; Pham, 2009).

A common way of handling continuous RL problems is to discretize continuous-time stochastic processes to fit with the discrete RL framework. However, discretization results in errors caused by partition states, actions, and times, and learning algorithms such as DQN and deep deterministic policy gradient collapse when time steps in the discretization decrease (Lillicrap et al., 2015; Tallec, Blier, and Ollivier, 2019).

Alternative approaches include working directly in the continuous-time framework. The direct approaches may be advantageous in that they avoid the issues encountered in the discretization approach. Continuous temporal-difference (TD) learning was introduced by Doya (2000) for a deterministic setting in which the underlying model is deterministic and governed by an ordinary differential equation. However, this approach is unstable

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

and may even diverge when applying to data generated from the deterministic model contaminated with noise. This paper considers a continuous-time stochastic diffusion model as a noisy version of a deterministic model. Because of noise, there are measurement errors in the observed data. We provide a new RL framework to handle the RL problem and propose a novel RL approach based on nonparametric kernel smoothing and temporal-difference to solve the RL problem with noisy data. We investigate its asymptotic properties and check its finite-sample behaviors. Our study demonstrates the good performance of the proposed method.

The rest of this paper is organized as follows. Section 2 introduces basic concepts and methodologies in RL. Section 3 describes continuous-time RL for an ODE model and continuous-time stochastic control for a SDE model and establishes their asymptotic relationship. Section 4 provides a new RL framework for an ODE model in which the RL learning problem is based on data generated from a SDE model, which can be viewed as a noisy version of the ODE model. We propose a novel methodology based on nonparametric smoothing and continuous-time TD to solve the continuous-time RL problem. Section 5 presents a numerical study on a pendulum-swinging RL task to check the performance of the proposed RL methodology. Section 6 features our concluding remarks. Finally, all the technical proofs are compiled in the Supplementary Materials.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

2. Brief review of reinforcement learning

2.1 MDP and dynamic programming

RL is usually described through an environment, an agent, and their interaction in terms of the Markov decision process (MDP). An MDP consists of a Markov system (environment) and a controller (agent). The system generates a state, and the controller produces an action according to the state and its associated reward. Next, the system generates another state based on the given state and action and the associated reward, and again the controller produces another action in response to the new state and its associated reward. The system and controller continue to interact and yield states and actions to maximize rewards, as illustrated in Figure 1.

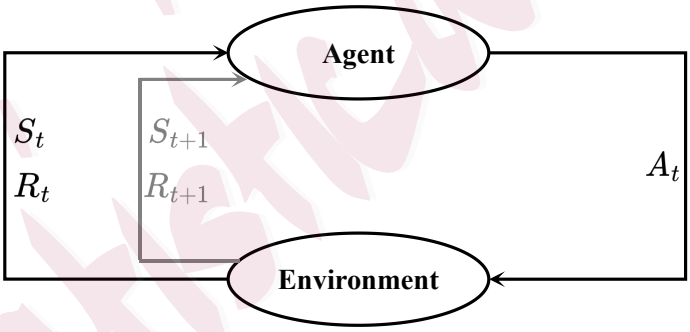


Figure 1: The agent-environment interaction in an MDP.

Consider an MDP system and denote its state space and action space by $\mathcal{S}$ and $\mathcal{A}$, respectively. Let $t = 0, 1, 2, \ldots$ be discrete time steps. Start at the initial state S_0 and action A_0. At each time step t, the controller receives a system state $S_t \in \mathcal{S}$ and responds by producing an action $A_t = \pi(S_t) \in \mathcal{A}$, where $\pi(\cdot)$ is called a policy that specifies the rule for the controller to choose action in response to a given state. A

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

reward $R_{t+1} = r(S_t, A_t)$ is calculated based on the given state S_t and action A_t, where $r(\cdot, \cdot)$ is called a reward function. Then, the system generates a new state S_{t+1}, and the controller produces a new action $A_{t+1} = \pi(S_{t+1})$ after receiving the new state S_{t+1} and the associated reward R_{t+1}. This cycle repeats until the process terminates. We call $S_0, A_0, S_1, R_1, A_1, S_2, \ldots,$ a trajectory or an episode. The RL goal is to learn an optimal policy to maximize a value function that is defined as an expected cumulated reward as follows:

$$V^\pi(s) = \mathbb{E}_\pi \left(\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \middle| S_0 = s \right),$$

where $\gamma \in (0, 1)$ is a constant called the discounted factor. Denote the optimal value function by $V(s) = \max_\pi V^\pi(s)$. An optimal policy is defined as a policy whose value function is equal to the optimal value function. The value function $V^\pi(s)$ satisfies the Bellman equation,

$$V^\pi(s) = \mathbb{E}_\pi[r(s, \pi(s)) + \gamma V^\pi(S_1) \mid S_0 = s], \tag{2.1}$$

and the optimal value function obeys the Bellman optimality equation,

$$V(s) = \sup_{a \in \mathcal{A}} \mathbb{E}[r(s, a) + \gamma V(S_1) \mid S_0 = s, A_0 = a]. \tag{2.2}$$

Dynamic programming algorithms are available to solve these equations to obtain the optimal value function and thus an optimal policy (Bellman, 1956; Bertsekas, 2017; Howard, 1960; Szepesvári, 2010).

2.2 RL approaches

Dynamic programming described in Section 2.1 requires the knowledge of the underlying MDP, like the Markov transition probability kernel for evaluating the optimal

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

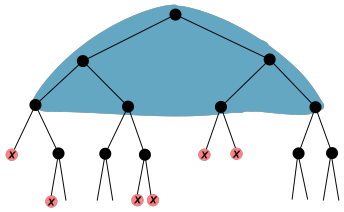
value function and finding an optimal policy. In the RL setup, we use observed data to estimate value function $V^{\pi}(\cdot)$, optimal value function $V(\cdot)$, and an optimal policy. Three basic RL approaches are dynamic programming, Monte-Carlo, and TD learning methods (Sutton and Barto, 2018; Szepesvári, 2010); Figure 2 provides an intuitive illustration of their similarities and differences. Dynamic programming methods take advantage of treating the Bellman equations as fixed point equations to develop greedy algorithms for learning a value function and estimating an optimal policy. However, it requires the entire state set of the MDP, rather than the sample experience used by Monte-Carlo and TD methods. Monte-Carlo methods estimate value function $V^{\pi}(s)$ based on n episodes, where an episode refers to a sample sequence of states, actions, and rewards from an actual or simulated interaction between the environment and agent under an underlying policy π. Denote the t-th episode by $E_t = (S_{1t}, A_{1t}, R_{2t}, S_{2t}, A_{2t}, \dots)$ and its corresponding total reward by G_t. Then, $V^{\pi}(s)$ is estimated by an average of G_t.

TD is widely regarded as a central and novel idea in RL. Denote by $\hat{V}(\cdot)$ the estimated value function, and define the TD(0) error as follows:

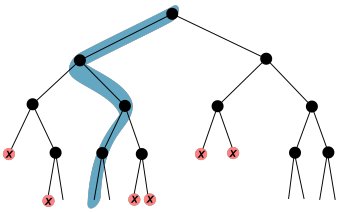
$$\delta_t = R_{t+1} + \gamma \hat{V}\left(S_{t+1}\right) - \hat{V}\left(S_t\right),$$

which is the difference between the estimated value function of the current state and its one-step ahead estimation using the estimated value function of the next state. The TD(0) algorithm uses the TD(0) error to update the estimated value function of the current state to obtain the following iterative updating rule for the finite state space

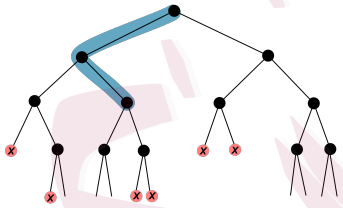
Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407



(a) Dynamic Programing



(b) Monte Carlo



(c) TD

Figure 2: Learning diagram in the Markov decision process. The roots are initial states of the Markov process. Nodes with a cross are the terminated states. The shaded areas represent states used in one learning update.

case:

$$\hat{V}(S_t) \leftarrow \hat{V}(S_t) + \eta \left[R_{t+1} + \gamma \hat{V}(S_{t+1}) - \hat{V}(S_t) \right], \tag{2.3}$$

where η is a step-size parameter. We may generalize the one-step ahead estimation used in TD(0) to the k-step estimation and obtain a corresponding so-called k-step TD method. Because TD methods involve the bootstrapping of MDP dynamic programming, such as Bellman equations, we refer to TD as a bootstrapping procedure in RL. TD methods update at every step but do not require episode experience. In contrast, Monte Carlo methods must wait till the end of an episode to have the final reward G_t so that we can form error $G_t - \hat{V}(S_t)$ to update the estimated value function of the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

current state and learn the value function.

TD(λ) provides a bridge to unify TD(0) and Monte Carlo methods. It is defined as a weighted sum of the k-step TD with weight $\lambda^{k-1}(1-\lambda)$, $k = 0, 1, 2, \cdots$, where λ is a parameter and belongs to $[0, 1]$. TD(λ) with $\lambda = 1$ corresponds to Monte Carlo methods, while TD(λ) with $\lambda = 0$ reduces to TD(0). The value of λ between 0 and 1 controls the weights of the k-step TD and produces intermediate methods between these two extreme methods. TD(λ) algorithms are usually described using eligibility trace. Assign e_t and w_t to denote the eligibility trace and weight vectors, respectively. Let $\hat{V}(S_t, w_t)$ be an estimated value function that depends on state S_t and weight w_t. The TD(λ) updating rule for the continuous state space case is given by

$$e_t \leftarrow \gamma\lambda e_{t-1} + \nabla\hat{V}(S_t, w_t), \tag{2.4}$$

$$\delta_t \leftarrow R_{t+1} + \gamma\hat{V}\left(S_{t+1}, w_t\right) - \hat{V}\left(S_t, w_t\right), \tag{2.5}$$

$$w_{t+1} \leftarrow w_t + \eta\delta_t e_t, \tag{2.6}$$

where the gradient ∇ in (2.4) is taken with respect to w_t, and η is a step-size parameter.

3. Continuous-time reinforcement learning

3.1 Continuous-time dynamic programing and RL

The continuous analog of MDP dynamic programming presented in Section 2.1 is established in stochastic control. Consider a diffusion processes governed by the following stochastic differential equation (SDE):

$$dX_t = b(X_t, \alpha_t)dt + \sigma(X_t, \alpha_t)dW_t, \tag{3.1}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

where α_t and W_t are, respectively, a control stochastic process and a Brownian motion defined on a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}\}_{t\geq 0}, P)$ that satisfies the usual conditions in stochastic calculus, and $b(\cdot,\cdot)$ and $\sigma(\cdot,\cdot)$ are drift and diffusion coefficients, respectively, which satisfy common Lipschitz and growth conditions to guarantee the existence and uniqueness of a solution to SDE (3.1) (Ikeda and Watanabe, 1989; Karatzas and Shreve, 1997; Kloeden and Platen, 1995; Pham, 2009).

Similar to the discrete case, denote by $r(\cdot,\cdot)$ a reward function and $\pi(\cdot)$ a policy with $\alpha_t = \pi(X_t)$. Define a value function

$$V^{\pi}(x) = E\left[\int_0^{\infty} e^{-\beta t} r(X_t, \alpha_t) dt \middle| X_0 = x\right], \tag{3.2}$$

where β is a constant often called the continuous discounted factor. The optimal value function is defined as $V(s) = \max_{\pi} V^{\pi}(s)$.

The continuous counterparts of the Bellman equations (2.1)-(2.2) in the discrete case are two partial differential equations for the continuous-time case (Recht, 2019; Powell and Ma, 2011; Pham, 2009; Kushner and Dupuis, 2001; Bertsekas, 2017), described as follows:

$$\beta V^{\pi}(x) = b(x,a)\frac{\partial V^{\pi}}{\partial x} + \frac{1}{2} tr\left(\sigma(x,a)\sigma'(x,a)\frac{\partial^2 V^{\pi}}{\partial x^2}\right) + r(x,a), \quad a = \pi(x), \tag{3.3}$$

and

$$\beta V(x) = \sup_a \left\{ b(x,a)\frac{\partial V}{\partial x} + \frac{1}{2} tr\left(\sigma(x,a)\sigma'(x,a)\frac{\partial^2 V}{\partial x^2}\right) + r(x,a)\right\}. \tag{3.4}$$

Equation (3.4) is called the Hamilton-Jacobi-Bellman (HJB) equations in the continuous-time stochastic control. Continuous-time RL has been studied for the SDE model (3.1), and RL approaches are investigated for the corresponding setup (Jia and Zhou, 2022a,b,c; Wang et al., 2020).

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

3.2 Continuous-time reinforcement learning in the deterministic case

Consider a deterministic continuous-time system with an underlying process to obey the following ordinary differential equation (ODE),

$$dX_t = b(X_t, \alpha_t)dt, \tag{3.5}$$

where α_t is a control process governed by a policy $\pi(\cdot)$ through $\alpha_t = \pi(X_t)$. ODE (3.5) corresponds to SDE (3.1) with $\sigma(\cdot, \cdot) = 0$. Given a value function $V(X_t)$ at time t, the continuous TD error is defined to be

$$\delta(t) = r(X_t, \pi(X_t)) + \dot{V}(X_t) - \beta V(X_t), \tag{3.6}$$

where $\dot{V}(X_t)$ denotes the derivative of $V(X_t)$ with respect to t.

Approximating the derivative $\dot{V}(X_t)$ in a continuous TD error by the backward Euler approximation scheme,

$$\dot{V}(X_t) \approx \frac{V(X_t) - V(X_{t-\Delta t})}{\Delta t},$$

we convert (3.6) into

$$\begin{aligned}
\delta(t) &\approx r(X_t, \pi(X_t)) + \frac{V(X_t) - V(X_{t-\Delta t})}{\Delta t} - \beta V(X_t) \\
&= r(X_t, \pi(X_t)) + \frac{1}{\Delta t}\left[(1 - \beta\Delta t)\, V(X_t) - V(X_{t-\Delta t})\right] \\
&\approx r(X_t, \pi(X_t)) + \frac{1}{\Delta t}\left[e^{-\beta\Delta t} V(X_t) - V(X_{t-\Delta t})\right],
\end{aligned} \tag{3.7}$$

which matches the discrete TD(0) error with time step Δt, except for a scaling factor $\frac{1}{\Delta t}$.

The value function learning in a continuous state space usually relies on some function approximation, such as linear combinations of features or weights. Denote by $w(t)$ a

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

feature or weight parameter and $\hat{V}(X_t, w(t))$ the current estimate of the value function based on $w(t)$. Given the continuous TD error $\delta(t)$, the continuous TD(λ) learning has the following eligibility trace $e(t)$, weight $w(t)$, and associated updating rules:

$$\dot{w}(t) = \eta\delta(t)e(t), \tag{3.8}$$

$$\delta(t) = r(X_t, \pi(X_t)) + \frac{d\hat{V}(X_t, w(t))}{dt} - \beta\hat{V}(X_t, w(t)), \tag{3.9}$$

$$\dot{e}(t) = -\frac{1}{\kappa}e(t) + \frac{\partial\hat{V}(X_t; w(t))}{\partial w(t)} \tag{3.10}$$

where $0 < \kappa \leq 1/\beta$ is the time constant of the eligibility trace (Doya, 2000).

With the estimated value function $\hat{V}(x)$ by the TD method, the HJB equation leads to the the following greedy action:

$$\pi(x) = \arg\max_{\mathbf{a}\in\mathcal{A}} \left[r(x, \mathbf{a}) + \frac{\partial\hat{V}(x)}{\partial x} b(x, \mathbf{a}) \right]. \tag{3.11}$$

Assume that

$$b(x, \mathbf{a}) \text{ is linear in } \mathbf{a}, \text{ and } \; r(x, \mathbf{a}) = r_0(x) - \sum_{j=1}^{d} r_j\left(a_j\right), \tag{3.12}$$

where $\mathbf{a} = (a_1, \cdots, a_d)^T$, $r_0(\cdot)$ is the reward for state x, and $r_1(a_1), \cdots, r_d(a_d)$ are cost functions for action variables $a_1, \cdots, a_d$. Suppose that r_j are convex, and their derivatives are denoted by $\dot{r}_j$. Then we have

$$\pi(x) = \left\{ \dot{r}_j^{-1}\left(\frac{\partial b(x, \mathbf{a})}{\partial a_j} \frac{\partial\hat{V}(x)}{\partial x} \right), j = 1, \cdots, d \right\}. \tag{3.13}$$

Furthermore, a small perturbation is put into the greedy action by adding noise in the function $\dot{r}_j^{-1}(\cdot)$ to increase exploration:

$$\pi(x) = \left\{ \dot{r}_j^{-1}\left(\frac{\partial b(x, \mathbf{a})}{\partial a_j} \frac{\partial\hat{V}(x)}{\partial x} + \text{noise} \right), j = 1, \cdots, d \right\}. \tag{3.14}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

3.3 Asymptotic analysis of optimal value functions and optimal policies under deterministic and stochastic models

We consider an RL problem for a deterministic model with noise contamination. Specifically, the problem is to learn the deterministic model (3.5) based on data generated from a form of the stochastic model (3.1). Continuous TD developed for deterministic processes cannot handle stochastic processes like (3.1), because the stochastic processes are not differentiable. Also, TD approaches based on the HJB optimality equation (3.4) require the knowledge of drift and diffusion coefficients, where the coefficients are hard to statistically estimate from discrete samples. This section establishes some relationships between the value functions for the deterministic and stochastic models. In the next section, we present a new framework for the RL problem and use nonparametric kernel smoothing to develop an RL method to solve the RL problem.

Denote $X_t(\sigma)$ as a solution to SDE (3.1). Here, the notation $X_t(\sigma)$ makes its dependence on σ explicitly, and $X_t(0)$ corresponds to a deterministic solution to ODE (3.5). The stochastic model (3.1) is considered as a noisy version of the deterministic model (3.5). We need the following technical assumptions to facilitate our analysis.

Assumptions

(A1) $\pi(x)$ and $\sigma(x, \pi(x))$ are bounded.

(A2) $b(x, a)$, $\sigma(x, a)$, and $\pi(x)$ are continuously differentiable and Lipschitz continuous with Lipschitz constants L_1, L_2, and M, respectively; that is, $|b(x_1, a_1) - b(x_2, a_2))| \leq L_1(|x_1 - x_2| + |a_1 - a_2|)$, and $|\sigma(x_1, a_1) - \sigma(x_2, a_2)| \leq L_2(|x_1 - x_2| + |a_1 - a_2|)$, and $|\pi(x_1) - \pi(x_2)| \leq M|x_1 - x_2|$.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

(A3) $\mathrm{Var}\left(X_t(\sigma)\right)$ and $\mathrm{Var}\left(b\left(X_t(\sigma), \pi(X_t(\sigma))\right)\right)$ as functions of t are bounded over t in any bounded interval.

(A4) The reward function $r(x, a)$ is bounded, continuously differentiable, and Lipschitz continuous with a Lipschitz constant L_r (i.e., $|r(x_1, a_1) - r(x_2, a_2))| \leq L_r[|x_1 - x_2| + |a_1 - a_2|]$).

The following theorem shows the closeness of the underlying processes between the deterministic and stochastic models when the diffusion coefficient is very small.

Theorem 1. *Assume assumptions (A1) - (A4), and suppose that σ can be decomposed as a product of two factors ε and ς; that is, $\sigma = \varepsilon\varsigma$, where ε is a constant, and ζ is bounded. We have*

$$\sup_{0\leq t\leq T, \pi\in Lip_M} \mathbb{E}\left[|X_t(\varepsilon\varsigma) - X_t(0)|^2 \mid X_0(\varepsilon\varsigma) = X_0(0) = x\right] \leq \varepsilon^2 M_T^x,$$

where Lip_M denotes the class of policies that are bounded by a constant and Lipschitz continuous with Lipschitz constant M, $g_\varepsilon(t, x) = \mathbb{E}\left[\int_0^t \varsigma^2(X_s(\varepsilon\varsigma), \pi(X_s(\varepsilon\varsigma)))ds \mid X_0(\varepsilon\varsigma) = x\right]$,

$$M_T^x = 2\left(g_\varepsilon(T, x) + 2L^2T\int_0^T g_\varepsilon(s, x)\exp\{2L^2T(T-s)\}ds\right),$$

and $L = \max\{L_1, L_2\}(M + 1)$. For the given T, both $g_\varepsilon(t, x)$ and M_T^x are bounded for $t \in [0, T]$ and all ε and x.

Remark 1. Theorem 1 indicates that as $\varepsilon \to 0$, the diffusion coefficient goes to zero, and the difference between the diffusion process $X_t(\sigma)$ and the deterministic process $X_t(0)$ converges in mean-square to zero. That is, $X_t(\sigma)$ and $X_t(0)$ have a negligible difference when σ is small enough. This implies that for very small σ, the effect of

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

random noise in SDE (3.1) is negligible, and the stochastic process $X_t(\sigma)$ generated from SDE (3.1) is close to the deterministic $X_t(0)$ generated from ODE (3.5). Thus, when σ is negligibly small, we may practically treat $X_t(\sigma)$ as $X_t(0)$, and the continuous policy evaluation and improvement developed for the deterministic model (3.5) can be effectively applied to the stochastic model (3.1).

In the next theorem, we show the closeness of value functions, optimal value functions, and optimal policies between the deterministic and stochastic cases when the diffusion coefficient is very small. Denote by $V_\sigma^\pi(x)$ and $V_0^\pi(x)$ the value functions under policy π for models (3.1) and (3.5), respectively, and define the corresponding optimal value functions as follows:

$$V_\sigma^{*,M}(x) = \sup_{\pi \in \mathrm{Lip}_M} V_\sigma^\pi(x), \quad V_0^{*,M}(x) = \sup_{\pi \in \mathrm{Lip}_M} V_0^\pi(x),$$

where Lip_M stands for the class of Lipschtiz-M policies given in Theorem 1. Denote by $\pi_\sigma^{*,M}(x)$ and $\pi_0^{*,M}(x)$ the optimal policies corresponding to $V_\sigma^{*,M}(x)$ and $V_0^{*,M}(x)$, respectively.

Theorem 2. *Assume assumptions (A1) - (A4), and $\sigma = \varepsilon\varsigma$ as in Theorem 1. We have that as $\varepsilon \to 0$, $V_{\varepsilon\varsigma}^\pi(x) \to V_0^\pi(x)$ uniformly over $\pi \in Lip_M$, and $V_{\varepsilon\varsigma}^{*,M}(x) \to V_0^{*,M}(x)$. Furthermore, as $\varepsilon \to 0$, any optimal policy $\pi_{\varepsilon\varsigma}^{*,M}(x)$ under the stochastic model (3.1) must converge to an optimal policy $\pi_0^{*,M}(x)$ under the deterministic model (3.5).*

Remark 2. Theorem 2 establishes the convergence of value functions, optimal value functions, and optimal policies for the stochastic model to the corresponding counterparts of the deterministic model when the diffusion coefficient of the stochastic

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

model goes to zero. It provides theoretical foundations for the proposed nonparametric smoothing-based RL approach, which will be presented in the next section.

4. Nonparametric smoothing-based RL approaches in the continuous-time stochastic setting

Consider a continuous-time RL problem where the underlying ideal model is the deterministic model (3.5), but observed samples are generated from a special form of the stochastic model (3.1), which is treated as a noisy version of the ideal deterministic model (3.5). The approach developed in Section 3 cannot be directly applied, because Theorem 2 requires a very small σ in model (3.1). Typical continuous-time RL problems with model (3.1) often do not have a negligibly small diffusion coefficient σ. Nonparametric smoothing comes to our rescue. Furthermore, because of noises, there are measurement errors in the observed data. Thus, we need a new RL framework to accommodate the RL problem.

4.1 A new RL framework for noisy data

Consider an RL problem where the true model to learn is the deterministic ODE model (3.5), and observed data are generated from some form of the SDE model (3.1), which is viewed as a noisy version of the deterministic ODE model. Because the true model is not directly observable, the observed data are sampled from a noisy version of the true model; namely, a form of the SDE model (3.1). The standard RL framework may not be suitable to accommodate RL learning with noisy data. Recall that the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

standard RL framework contains a system and an agent. The system produces a state, and the agent generates an action according to the state and evaluates its associated reward. Next, the system produces another state based on the given state, action, and the associated reward. The agent then generates another action in response to the new state and evaluates its associated reward. The system and agent continue to interact and produce states and actions to maximize rewards. Applying the standard RL framework to the described noisy RL problem results in the usual RL problem and its associated solution for the SDE model (3.1) (Bertsekas, 2017; Jia and Zhou, 2022a,b,c; Pham, 2009; Wang et al., 2020). In our case, the model (3.1) is simply a noisy version of the true model (3.5). An RL solution to the model (3.1) can be heavily influenced by the stochastic component driven by a Brownian motion, which is considered as noise in our case and thus provides a spurious solution to the described RL problem with the true model (3.5). The phenomenon has been widely investigated in machine learning, such as classification for noisy data Friedman et al. (2001). For the purpose of simple illustration, consider an RL problem for the finite discrete MDP model described in Section 2.1 where the true states of the MDP are not observable, and the system generates noisy states of the MDP—states of a corresponding hidden Markov model. Just as the control and RL solutions for models (3.5) and (3.1) differ, so do the solutions for the Markov and hidden Markov models (Elliott et al., 2008; Szepesvári, 2010). For the RL learning with noisy data, because the agent must have some proxy of the true states to generate actions and evaluate rewords, we need de-noising or error removing in the RL framework to recover or estimate the true states from the noisy

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

state data so that the interaction of the system and agent can be carried out. Since statistical procedures of de-noising or error removing require multiple observations (Yi, 2017; Cappé et al., 2005; Elliott et al., 2008), at each interaction of the system and agent, the system needs to generate multiple noisy states to estimate the true states. Then, the agent produces an action according to the last estimated state and its associated reward. The next section describes the RL problem with the true model (3.5) and noisy data generated from the model (3.1).

4.2 An RL approach based on nonparametric smoothing

Given the data $X_t(\sigma)$ observed from (3.1), we apply a kernel method to smooth the data $X_t(\sigma)$ and use the smoothed data to estimate $X_t(0)$ from (3.5). Then, we apply the smoothed data to the continuous-time RL methods, such as the TD developed for the deterministic model (3.5). The proposed approach is based on Theorems 1 and 2 and the fact that smoothing reduces the random effect (noises or errors) and can make the smoothed $X_t(\sigma)$ close to $X_t(0)$. Thus, the value functions and policies for the smoothed $X_t(\sigma)$ are close to those for $X_t(0)$.

Given an estimated state at current time t, evaluate the policy $\pi(\cdot)$ at the current estimated state to produce an action α_t. Then, use the same action α_t to generate m states from (3.1) and denote them by $X_{t_i}(\sigma)$, where $t_i \in [t, t+\Delta t]$, $m = \#\{t_i : t \leq t_i \leq t+\Delta t\}$, and Δt denotes the length of the time window. Let $K(x)$ be a kernel function with support on $[0, 1]$. We apply kernel smoothing to $X_{t_i}(\sigma)$ to obtain a kernel estimator, denoted by $\tilde{X}_t^h(\sigma)$, as illustrated in Figure 3, where h denotes the bandwidth

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

in the kernel smoothing (Friedman et al., 2001). Specially, we define

$$\overline{\Delta X_t^h(\sigma)} := \frac{1}{m}\sum_{i=1}^{m} K\left(\frac{t_i - t}{h}\right)\left[X_{t_i}(\sigma) - X_t(\sigma)\right]. \tag{4.1}$$

The new process $\tilde{X}_t^h(\sigma)$ is obtained by the following procedure: Given the current estimated state $\tilde{X}_t^h(\sigma)$, we estimate the state at the next step $t + \Delta t$ by

$$\tilde{X}_{t+\Delta t}^h(\sigma) = \tilde{X}_t^h(\sigma) + \overline{\Delta X_t^h(\sigma)}. \tag{4.2}$$

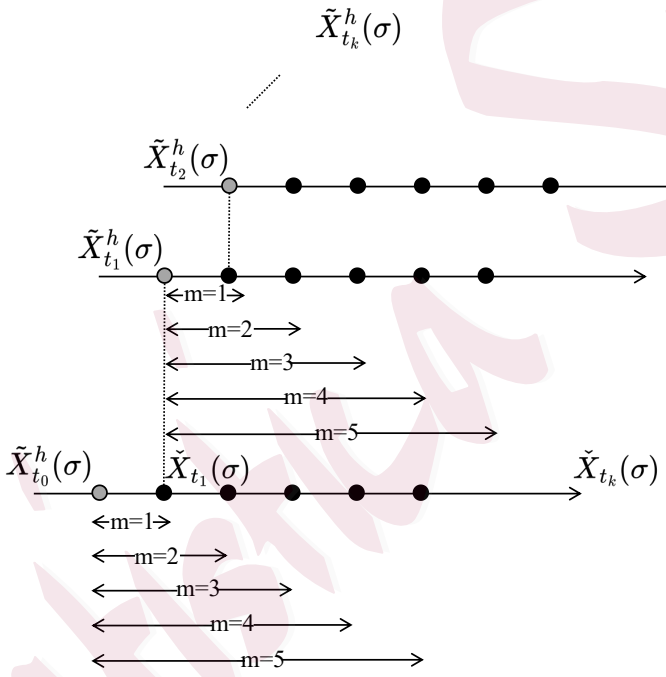


Figure 3: Smoothing scheme with bandwidth h.

The framework and interaction setup is summarized by a pseudo code in Table 1 and described as follows. Given the current estimated state $\tilde{X}_t^h(\sigma)$, we can produce an action $\alpha_t = \pi(\tilde{X}_t^h(\sigma))$, evaluate the associated reward $r(\tilde{X}_t^h(\sigma), \pi(\tilde{X}_t^h(\sigma)))$, and then use the same action α_t to generate m states $X_{t_i}(\sigma)$, $t_i \in [t, t + \Delta t]$, $i = 1, \cdots, m$, from the SDE model (3.1). We then compute the next state estimator $\tilde{X}_{t+\Delta t}^h(\sigma)$ by

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

Table 1: Pseudo code for RL with noisy data

Input: an initial state, policy $\pi(\cdot)$, and reward $r(\cdot,\cdot)$.
Step 1. Given the current estimated state $\tilde{X}_t^h(\sigma)$, compute $\alpha_t = \pi(\tilde{X}_t^h(\sigma))$ and $r(\tilde{X}_t^h(\sigma), \pi(\tilde{X}_t^h(\sigma)))$; use α_t to generate m states $X_{t_i}(\sigma)$, $t_i \in [t, t+\Delta t]$, from (3.1).
Step 2. Calculate the next state estimator $\tilde{X}_{t+\Delta t}^h(\sigma)$ by (4.1)-(4.2), and compute $\alpha_{t+\Delta t} = \pi(\tilde{X}_{t+\Delta t}^h(\sigma))$ and $r(\tilde{X}_{t+\Delta t}^h(\sigma), \pi(\tilde{X}_{t+\Delta t}^h(\sigma)))$; use $\alpha_{t+\Delta t}$ to generate m states $X_{t_i}(\sigma)$, $t_i \in [t+\Delta t, t+2\Delta t]$, from (3.1).
Step 3. Repeat Step 1 and Step 2.

(4.1)-(4.2), produce an action $\alpha_{t+\Delta t} = \pi(\tilde{X}_{t+\Delta t}^h(\sigma))$, evaluate the associated reward $r(\tilde{X}_{t+\Delta t}^h(\sigma), \pi(\tilde{X}_{t+\Delta t}^h(\sigma)))$, and then use the same action $\alpha_{t+\Delta t}$ to generate m states $X_{t_i}(\sigma)$, $t_i \in [t+\Delta t, t+2\Delta t]$, $i = 1, \cdots, m$, from the SDE model (3.1). The interaction process continues and the proposed method yields estimated states and actions to maximize rewards.

Nonparametric smoothing indicates that as $\Delta t \to 0$, $h \to 0$, $m \to \infty$, and $mh \to 0$, the estimated state $\tilde{X}_t^h(\sigma)$ is a smoothed process with a negligible noise component, and thus $\tilde{X}_t^h(\sigma)$ is close to the deterministic solution $X_t(0)$ of ODE (3.5). Hence, the generated action $\pi(\tilde{X}_t^h(\sigma))$ and the associated reward $r(\tilde{X}_t^h(\sigma), \pi(\tilde{X}_t^h(\sigma)))$ are approximately equal to their true counterparts $\pi(X_t(0))$ and $r(X_t(0), \pi(X_t(0)))$, respectively. Consequently, we can compute the TD error $\delta(t)$ from (3.6) and carry out the policy evaluation and improvement by the TD learning and greedy algorithms from (3.8)-(3.11).

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

4.3 Procedure to implement RL based on nonparametric smoothing

We apply the 4th order Runge-Kutta (RK) method to solve the ODE given below. For any $T > 0$, consider the ODE

$$\dot{X}_t = f(X_t, \varpi(X_t), t),\ t \in [0, T], \tag{4.3}$$

and obtain an iterative algorithm as follows:

$$Y_{t_\ell} = Y_{t_{\ell-1}} + \frac{1}{6}\left(\Upsilon_1^{\ell-1} + 2\Upsilon_2^{\ell-1} + 2\Upsilon_3^{\ell-1} + \Upsilon_4^{\ell-1}\right),\ \ell = 1, \cdots, N, \tag{4.4}$$

where Y_{t_0} is an initial value,

$$\begin{cases} \Upsilon_1^{\ell-1} = (T/N) f\left(Y_{t_{\ell-1}}, \varpi(Y_{t_{\ell-1}}), t_{\ell-1}\right), \\ \Upsilon_2^{\ell-1} = (T/N) f\left(Y_{t_{\ell-1}} + \Upsilon_1^{\ell-1}/2, \varpi(Y_{t_{\ell-1}}), t_{\ell-1} + T/(2N)\right), \\ \Upsilon_3^{\ell-1} = (T/N) f\left(Y_{t_{\ell-1}} + \Upsilon_2^{\ell-1}/2, \varpi(Y_{t_{\ell-1}}), t_{\ell-1} + T/(2N)\right), \\ \Upsilon_4^{\ell-1} = (T/N) f\left(Y_{t_{\ell-1}} + \Upsilon_3^{\ell-1}, \varpi(Y_{t_{\ell-1}}), t_{\ell-1} + T/N\right), \end{cases} \tag{4.5}$$

f and ϖ are appropriate functions, N denotes the number of iterations, and $t_\ell = \ell T/N$ with step size T/N. That is, we recursively apply the algorithm (4.4)-(4.5) over time interval $[0, T]$, and the resultant N iterations provide the numerical solution of (4.3) at N time points t_ℓ.

We take the function $f(X_t, \varpi(X_t), t)$ in (4.3) to be $b(X_t, \alpha_t)$ for the deterministic case of ODE (3.5), and denote the resulting discrete process by $\check{X}_{t_\ell}(0)$. In the stochastic case of SDE (3.1), we proceed as follows. Denote by $\dot{W}_t$ the white noise (i.e., the derivative of Brownian motion W_t in the generalized function sense). We take the function $f(X_t, \varpi(X_t), t)$ in (4.3) to be $b(X_t, \alpha_t) + \sigma(X_t, \alpha_t)\dot{W}_t$ and use the 4th order

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

RK method (4.4)-(4.5) to generate discrete samples and apply kernel smoothing (4.1) and (4.2) to the generated discrete samples. We denote the resulting discrete smoothed samples by $\tilde{X}^h_{t_k}(\sigma)$ at discrete points $t_k = kT/n = kmT/N$, $k = 1, \cdots, n = N/m$. Specifically, the discrete smoothed samples are iteratively defined as follows:

$$\begin{aligned}\tilde{X}^h_{t_k}(\sigma) &= \tilde{X}^h_{t_{k-1}}(\sigma) + \overline{\Delta X^h_{t_{k-1}}(\sigma)} \\ &= \tilde{X}^h_{t_{k-1}}(\sigma) + \frac{1}{m}\sum_{j=1}^{m} K\left(\frac{t_{k-1+j} - t_{k-1}}{h}\right)\left[\check{X}_{t_{k-1+j}}(\sigma) - \tilde{X}^h_{t_{k-1}}(\sigma)\right],\end{aligned} \tag{4.6}$$

where states $\check{X}_{t_{k-1+j}}(\sigma)$, $j = 1, \cdots, m$, are assumed to be generated from the same action $\alpha_{t_{k-1}} = \pi(\tilde{X}^h_{t_{k-1}}(\sigma))$ using the algorithm (4.4)-(4.5) with $\varpi(\tilde{X}^h_{t_{k-1+j}}) = \pi(\tilde{X}^h_{t_{k-1}}(\sigma))$, and

$$\begin{cases} \check{X}_{t_k}(\sigma) = \tilde{X}^h_{t_{k-1}}(\sigma) + \frac{1}{6}\left(\Upsilon^{k-1,0}_{1,\sigma} + 2\Upsilon^{k-1,0}_{2,\sigma} + 2\Upsilon^{k-1,0}_{3,\sigma} + \Upsilon^{k-1,0}_{4,\sigma}\right), \\ \check{X}_{t_{k+1}}(\sigma) = \check{X}_{t_k}(\sigma) + \frac{1}{6}\left(\Upsilon^{k-1,1}_{1,\sigma} + 2\Upsilon^{k-1,1}_{2,\sigma} + 2\Upsilon^{k-1,1}_{3,\sigma} + \Upsilon^{k-1,1}_{4,\sigma}\right), \\ \vdots \\ \tilde{X}_{t_{k-1+m}}(\sigma) = \check{X}_{t_{k-1+m-1}}(\sigma) \\ \qquad\qquad + \frac{1}{6}\left(\Upsilon^{k-1,m-1}_{1,\sigma} + 2\Upsilon^{k-1,m-1}_{2,\sigma} + 2\Upsilon^{k-1,m-1}_{3,\sigma} + \Upsilon^{k-1,m-1}_{4,\sigma}\right), \end{cases} \tag{4.7}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

where

$$\begin{cases} \Upsilon_{1,\sigma}^{k-1,i} = \frac{T}{N} b\left(\check{X}_{t_{k-1+i}}(\sigma), \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right) \\ \quad + \frac{T}{N}\sigma\left(\check{X}_{t_{k-1+i}}(\sigma), \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right)\check{W}_i(t_{k-1}) \\ \Upsilon_{2,\sigma}^{k-1,i} = \frac{T}{N} b\left(\check{X}_{t_{k-1+i}}(\sigma) + \Upsilon_{1,\sigma}^{k-1,i}/2, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right) \\ \quad + \frac{T}{N}\sigma\left(\check{X}_{t_{k-1+i}}(\sigma) + \Upsilon_{1,\sigma}^{k-1,i}/2, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right)\check{W}_i(t_{k-1} + T/(2N)) \\ \Upsilon_{3,\sigma}^{k-1,i} = \frac{T}{N} b\left(\check{X}_{t_{k-1+i}}(\sigma) + \Upsilon_{2,\sigma}^{k-1,i}/2, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right) \\ \quad + \frac{T}{N}\sigma\left(\check{X}_{t_{k-1+i}}(\sigma) + \Upsilon_{2,\sigma}^{k-1,i}/2, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right)\check{W}_i(t_{k-1} + T/(2N)) \\ \Upsilon_{4,\sigma}^{k-1,i} = \frac{T}{N} b\left(\check{X}_{t_{k-1+i}}(\sigma) + \Upsilon_{3,\sigma}^{k-1,i}, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right) \\ \quad + \frac{T}{N}\sigma\left(\check{X}_{t_{k-1+i}}(\sigma) + \Upsilon_{3,\sigma}^{k-1,i}, \pi(\tilde{X}_{t_{k-1}}^h(\sigma))\right)\check{W}_i(t_{k-1} + T/N) \end{cases} \tag{4.8}$$

for $i = 0, 1, \ldots, m-1$, and $\check{W}_i$ in (4.8) are discrete white noises corresponding to $\dot{W}$ and independent of $\tilde{X}_{t_{k-1}}^h(\sigma)$. If the kernel function K is chosen to be a constant function, we obtain

$$\begin{aligned} \tilde{X}_{t_k}^h(\sigma) &= \tilde{X}_{t_{k-1}}^h(\sigma) + \frac{1}{m}\left[\check{X}_{t_{k-1+m}}(\sigma) - \tilde{X}_{\sigma}^h(t_{k-1})\right] \\ &= \tilde{X}_{t_{k-1}}^h(\sigma) + \frac{1}{6m}\sum_{i=0}^{m-1}\left(\Upsilon_{1,\sigma}^{k-1,i} + 2\Upsilon_{2,\sigma}^{k-1,i} + 2\Upsilon_{3,\sigma}^{k-1,i} + \Upsilon_{4,\sigma}^{k-1,i}\right). \end{aligned} \tag{4.9}$$

Given $\tilde{X}_{t_k}^h(\sigma)$, we compute the associated reward and the TD error by (3.6)-(3.7), estimate the corresponding value function, and perform the TD learning using (3.8)-(3.11).

4.4 Asymptotic theory

We fix notations to facilitate the development of the asymptotic theory. We fix the notations for value functions under deterministic and stochastic cases as follows:

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

1. $V_0^{\pi}(x)$ denotes the value function for the deterministic $X_t(0)$; namely, $V_0^{\pi}(x) = \int_0^{\infty} e^{-\beta t} r(X_t(0), \pi(X_t(0)))dt$, given $X_0(0) = x$. Let $V_0^{*,M}(x)$ be the corresponding optimal value function that is the maximum of $V_0^{\pi}(x)$ over $\pi \in \text{Lip}_M$. Denote by $\pi_0^{*,M}(x)$ an optimal policy, that is, $\pi_0^{*,M}(x) = \arg\max_{\pi \in \text{Lip}_M} V_0^{\pi}(x)$.

2. $V_{\sigma}^{\pi}(x)$ denotes the value function for the stochastic $X_t(\sigma)$; namely, $V_{\sigma}^{\pi}(x) = E\left[\int_0^{\infty} e^{-\beta t} r(X_t(\sigma), \pi(X_t(\sigma))dt \middle| X_0(\sigma) = x\right]$. Let $V_{\sigma}^{*,M}(x)$ be the corresponding optimal value function that is the maximum of $V_{\sigma}^{\pi}(x)$ over $\pi \in \text{Lip}_M$. Denote by $\pi_{\sigma}^{*,M}(x)$ an optimal policy for $X_t(\sigma)$; that is, $\pi_{\sigma}^{*,M}(x) = \arg\max_{\pi \in \text{Lip}_M} V_{\sigma}^{\pi}(x)$.

We specify the continuous-time process, value function, optimal value function, and optimal policy for the smoothed samples $\tilde{X}_{t_k}^h(\sigma)$ as follows.

1. Define a continuous-time process by interpolating $\tilde{X}_{t_k}^h(\sigma)$, and denote by $\tilde{X}_t^h(\sigma)$ the resulting continuous-time smoothed process.

2. Let $\tilde{V}_{\sigma}^{h,\pi}(x) = E\left[\int_0^{\infty} e^{-\beta t} r(\tilde{X}_t^h(\sigma), \pi(\tilde{X}_t^h(\sigma)))dt \middle| \tilde{X}_0^h(\sigma) = x\right]$ be the value function for $\tilde{X}_t^h(\sigma)$.

3. Let $\tilde{V}_{\sigma}^{h,*,M}(x) = \sup_{\pi \in \text{Lip}_M} \tilde{V}_{\sigma}^{h,\pi}(x)$ be the optimal value function for $\tilde{X}_t^h(\sigma)$.

4. Denote by $\tilde{\pi}_{\sigma}^{h,*,M}(x)$ an optimal policy for $\tilde{X}_t^h(\sigma)$; namely, $\tilde{\pi}_{\sigma}^{h,*,M}(x) = \arg\max_{\pi \in \text{Lip}_M} \tilde{V}_{\sigma}^{h,\pi}(x)$.

We have the following theorem to demonstrate that the smoothed samples $\tilde{X}_{t_k}^h$ are close to the deterministic samples $X_{t_k}(0)$.

Theorem 3. *Under assumptions (A1)-(A3), we have that for each x, as $h \to 0$, $m \to$*

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

∞ *and* $mh \to 0$,

$$\sup_{t_k = kT/n, 1 \le k \le n} \mathbb{E}\left[\left\{ \tilde{X}^h_{t_k}(\sigma) - X_{t_k}(0) \right\}^2 \middle| X_0(\sigma) = X_0(0) = x \right] = O\left(m^{-1}\right),$$

where the asymptotic notation $O(m^{-1})$ *means a bound* Cm^{-1} *for constant* C *that may depend on* x *but is free of* (h, m, n).

Remark 3. Theorem 3 shows that the mean square error between the smoothed samples $\tilde{X}^h_{t_k}(\sigma)$ defined by (4.6)-(4.8) and the deterministic samples $X_{t_k}(0)$ converges to 0 uniformly over $t_k = kT/n$, $k = 1, \cdots, n = N/m$.

The following theorem establishes the convergence of the value function, the optimal value function, and the optimal policy for the smoothed process $\tilde{X}^h_t(\sigma)$ to the corresponding counterparts of the deterministic $X_t(0)$.

Theorem 4. *Under assumptions (A1)-(A4), we have that as* $h \to 0$, $m \to \infty$ *and* $mh \to 0$,

$$\sup_{\pi \in Lip_M} |\tilde{V}^{h,\pi}_\sigma(x) - V^\pi_0(x)| \to 0,$$

and

$$\tilde{V}^{h,*,M}_\sigma(x) \to V^{*,M}_0(x).$$

Furthermore, as $h \to 0$, $m \to \infty$ *and* $mh \to 0$, *any optimal policy* $\tilde{\pi}^{h,*,M}_\sigma(x)$ *under the stochastic model (3.1) must converge to an optimal policy* $\pi^{*,M}_0(x)$ *under the deterministic model (3.5).*

Remark 4. Theorem 4 indicates that the value function, the optimal value function, and the optimal policy for the smoothed process $\tilde{X}^h_t(\sigma)$ converge to the corresponding targets for the deterministic $X_t(0)$.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

Remark 5. We consider a continuous-time RL task where the underlying ideal model is the deterministic model (3.5), but data samples are generated from the stochastic model (3.1), which is treated as a noisy version of the ideal deterministic model (3.5). Using Theorems 3 and 4, the proposed approach can obtain smoothed samples close to the deterministic samples from the ideal model, and the value functions, optimal value functions, and optimal polices for the smoothes samples converge to the corresponding counterparts for the deterministic samples. Thus, Theorems 3 and 4 together may validate the proposed RL approach by applying the continuous-time deterministic RL methods, such as TD, to the smoothed samples $\tilde{X}^h_{t_k}(\sigma)$ to render good solutions for the stochastic RL task.

5. A numerical study on a real RL task

We carried out a numerical study to check the performance of the proposed method. The study is about the continuous-time RL task of a pendulum swinging upwards with limited torque illustrated in Figure 4.

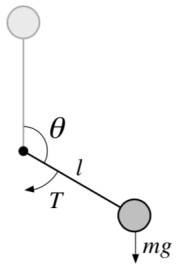


Figure 4: Pendulum

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

5.1 The RL setup

Denote the state variable of the pendulum by $X_t(\sigma) = (\theta_t, \omega_t)$, where θ_t represents the angle at time t between the pendulum position and the vertical up direction, with ω_t as its derivative. θ_t and ω_t are assumed to obey the following SDE:

$$\begin{cases} d\theta_t = \omega_t dt, \ t \in [0, T], \\ d\omega_t = \frac{1}{\mathcal{M}\mathcal{L}^2}(-\mu\,\omega_t + \mathcal{M}g\mathcal{L}\sin\theta_t + \alpha)dt + \sigma dW_t, \end{cases} \tag{5.1}$$

where W_t is a standard Brownian motion, σ is a constant to represent the noise level in the problem, α denotes the control variable with $\alpha_{max} = 5.0$, and $(\mathcal{M}, \mathcal{L}, g, \mu)$ are physical constants with assigned values $\mathcal{M} = \mathcal{L} = 1$, $g = 9.8$, and $\mu = 0.01$. When $\sigma = 0$, the model (5.1) reduces to the deterministic physics model of the pendulum in the ideal setting described by Doya (2000), and the noise model (5.1) is used to describe the pendulum behavior in a practical situation. To make the pendulum swinging upwards, the agent needs to swing it several times to build up momentum, and then decelerate it early enough to prevent the pendulum from falling over. The task is non-trivial when the maximal output torque α_{max} is smaller than the maximal load torque $\mathcal{L}\mathcal{M}g$. In our simulation study, we take $T = 20$ seconds and a step size equal to 0.002 in the 4th order RK method to simulate the process $X_t(\sigma)$.

We start every trial in an initial state $X_0(\sigma) = (\theta_0, \omega_0)$, where $\omega_0 = 0$, and θ_0 is chosen at random from the interval $(-\pi, \pi]$. Each trial is carried out to last for 20 seconds, unless the pendulum is over-rotated to the degree that the angle exceeds 5π. Upon such a failure, we terminate the trial and assign a reward $r(\theta, a) = -1$ for one second. We define a measure of the swing-up performance by the time, t_{up}, in which the pendulum

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

stays up at an angle between $-\pi/4$ and $\pi/4$. We consider a trial a success if t_{up} exceeds 10 seconds. Following (Doya, 2000) we adopt the reward function

$$r(\theta, a) = \cos(\theta) - 0.1\,\alpha_{\max}\left(\frac{4}{\pi^2}\right)\log\left(\cos\left(\frac{\pi a}{2\alpha_{\max}}\right)\right),$$

and the policy

$$\pi(X_t) = \frac{2}{\pi}\alpha_{\max}\arctan\left(\frac{5\pi}{\mathcal{ML}^2}\frac{\partial V(X_t)}{\partial X_t} + 0.5\min\{1, \max[0, 1 - V(X_t)]\}\,\Xi_t\right),$$

where Ξ_t is a noise process satisfying $\frac{d\Xi_t}{dt} = -\Xi_t + \varpi_t$, and ϖ_t are independent Gaussian random variables with zero mean and unit variance, and we model the value function by

$$V(x) = \sum_{k=1}^{225}\nu_{1k}b_k(x), \quad b_k(x) = \frac{e^{-\|\nu_{2k}(x-\nu_{3k})\|}}{\sum_{j=1}^{225}e^{-\|\nu_{2j}(x-\nu_{3j})\|}},$$

where $(\nu_{1k}, \nu_{2k}, \nu_{3k})$ are parameters. See (Doya, 2000) for details about the specifications of the reward function, policy, and value function.

5.2 Numerical results

Discrete smoothing samples are simulated from the model (5.1) using the 4th order RK method with $T = 20$, a step size equal to 0.002, and σ between 5 and 11, and by the proposed smoothing method. Denote the smoothed samples by $\tilde{X}^h_{t_k}$, $k = 1, \cdots, n$, where the step size is equal to T/N, $n = N/m$; m denotes the number of observations in the smoothing window, the smoothing bandwidth is given by $h = mT/N$, and the kernel function is chosen to be a constant function. Thus, $N = T/0.002 = 10,000$, and $n = 10,000/m$. We select m as from 1 to 5 and the bandwidth as $h = 0.002m$. We apply the continuous TD(λ) algorithm to the smoothed samples $\tilde{X}^h_{t_k}$ and test the performance

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

Table 2: Experiments settings with 4th order RK method.

step size	σ	m	Method
0.002	5, 6, 7, 8, 9, 10, 11	1	continuous TD
		2	smoothing & continuous TD with bandwidth h = 1 step size
		3	smoothing & continuous TD with bandwidth h = 2 step size
		4	smoothing & continuous TD with bandwidth h = 3 step size
		5	smoothing & continuous TD with bandwidth h = 4 step size

under various scenarios, as illustrated in Table 2. In each testing experiment, 100 trials of 20-second runs are used to train the TD algorithm. To increase exploration, we add in (3.14) a normal noise with mean zero and standard deviation 0.5. To account for the randomness of the training process, each test setting is repeated 500 times. Figure 5 shows the best performance for each 500 repetitions. From Figure 5 (a)-(d), we can observe that as the noise level increases, the deterministic continuous RL methods start to deteriorate and eventually fail completely. In contrast, the proposed approach provides good solutions to the RL problem. The numerical results indicate that the existing deterministic continuous RL methods do not work for the RL problem in the stochastic setting, and the proposed continuous RL approach based on nonparametric smoothing can solve the stochastic RL problem. We also compare the proposed approach with relevant existing algorithms, namely the online actor-critic (OAC) algorithm for the continuous-time case (Jia and Zhou, 2022b) and a discretized version of the continuous-time actor-critic (CAC) algorithm (Doya, 2000). We display in Figure

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

5 (e)-(f) the obtained results for the proposed method in comparison with the OAC and CAC algorithms. The results imply the better performance of our method over the existing algorithms. The findings confirm the established theory for the proposed RL methodology and that the existing methods are unable to handle RL problems with noisy data.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

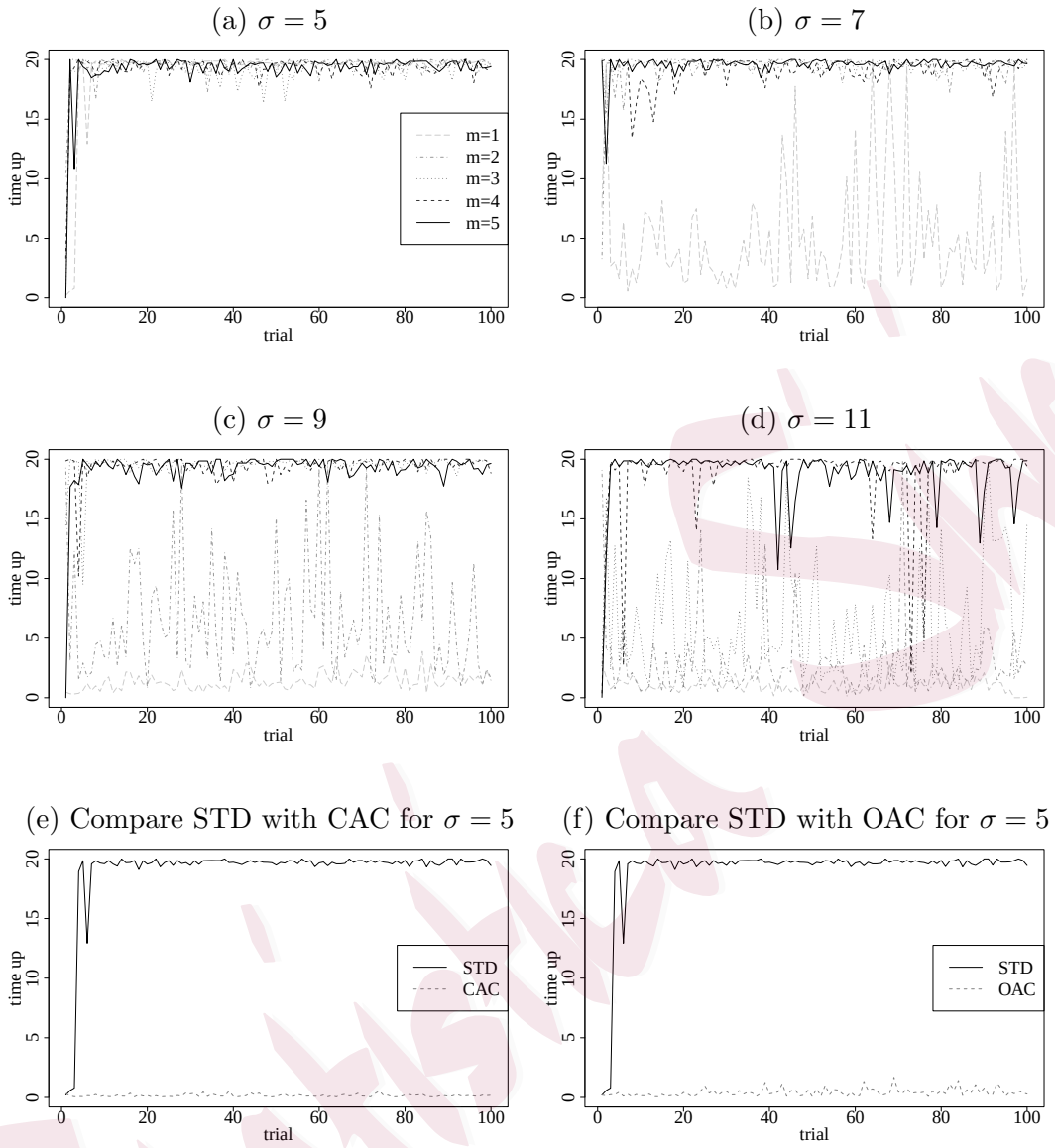


Figure 5: Plot of the stay-up time t_{up} against trials for solving the noisy pendulum swing-up RL problem by the existing and proposed methods under various combinations of bandwidth $h = 0.002m$ and noise level σ, where $m = 1, 2, 3, 4, 5$ correspond to curves in (a)-(d) with five different line types, $\sigma = 5, 7, 9, 11$ are for plots (a)-(d), and plots (e) and (f) correspond to curves for the comparison of the proposed smoothed TD (STD) method with the OAC and CAC algorithms, respectively.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

6. Concluding remarks

In this paper, we develop a new learning approach based on temporal-difference and nonparametric smoothing to solve reinforcement learning problems in a continuous-time stochastic setting, where the underlying ideal model is governed by an ODE, and data samples are generated from a SDE that is considered as a noisy version of the ODE model. Reinforcement learning methods have been widely developed for discrete-time MDP, but few RL techniques available for continuous-time stochastic processes. For example, continuous-time temporal-difference learning was developed for deterministic ODE models, but it is not applicable to stochastic models. In fact, the existing continuous-time temporal-difference learning method is unstable and diverges when applied to data generated from stochastic models. Furthermore, no RL techniques have been developed to handle RL problems with noisy data. We developed an approach based on nonparametric smoothing to handle the described continuous-time stochastic RL tasks with noisy data. We established an asymptotic theory for the proposed approach and conducted a numerical study to solve a pendulum RL task and check the finite sample performance of the proposed method. The theoretical analysis and numerical study show that the proposed learning approach delivers a robust performance. There are extensive research works in the literature on stochastic control for continuous-time stochastic processes governed by SDEs. However, there is a lack of good RL approaches for these continuous-time stochastic models, particularly the statistical aspect of RL. Some recent attempts have been made in this direction (Wang et al., 2020; Jia and Zhou, 2022a,b,c). Despite the extensive research on RL for discrete MDP, little

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

work has been done on RL with noisy data. We hope that our paper will contribute to the recent RL works to stimulate further research on continuous-time RL and RL with noisy data.

Supplementary Materials

The Supplementary Materials include the proofs of Theorem 1-Theorem 4.

Acknowledgments

The research of Yazhen Wang was supported in part by NSF grant DMS-1913149.

References

Bellman, R. (1956). Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, 42(10):767–769.

Bertsekas, D. P. (2017). *Dynamic Programming and Optimal Control (2 vol set, fourth edition)*. Athena Scientific.

Cappé, O., Moulines, E., and Ryden, T. (2005). *Inference in Hidden Markov Models*. Springer.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural computation*, 12(1):219–245.

Elliott, R. J., Aggoun, L., and Moore, J. B. (2008). *Hidden Markov Models: Estimation and Control*, volume 29. Springer, second edition.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer series in statistics New York, second edition.

Howard, R. (1960). *Dynamic Programming and Markov Processes*. Technology Press of Massachusetts Institute of Technology.

Ikeda, N. and Watanabe, S. (1989). *Stochastic Differential Equations and Diffusion Processes*, volume 24. North-Holland, second edition.

Jia, Y. and Zhou, X. (2022a). Policy evaluation and temporal-difference learning in continuous time and space: a martingale approach. *Journal of Machine Learning Research*, 23:1–55.

Jia, Y. and Zhou, X. (2022b). Policy gradient and actor-critic learning in continuous time and space: theory and algorithms. *Journal of Machine Learning Research*, 23:1–50.

Jia, Y. and Zhou, X. (2022c). q-learning in continuous time. *arXiv:2207.00713v1*.

Karatzas, I. and Shreve, S. E. (1997). *Brownian Motion and Stochastic Calculus*, volume 113. Springer, second edition.

Kloeden, P. E. and Platen, E. (1995). *Numerical Solution of Stochastic Differential Equations*, volume 23. Springer.

Kushner, H. J. and Dupuis, P. (2001). *Numerical Methods for Stochastic Control Problems in Continuous Time*, volume 24. Springer, second edition.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Pham, H. (2009). *Continuous-time Stochastic Control and Optimization with Financial Applications*, volume 61. Springer.

Powell, W. B. and Ma, J. (2011). A review of stochastic algorithms with continuous value function approximation and some new approximate policy iteration algorithms for multidimensional continuous applications. *Journal of Control Theory and Applications*, 9(3):336–352.

Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*, volume 135. MIT press Cambridge, second edition.

Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Morgan & Claypool.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0407

Tallec, C., Blier, L., and Ollivier, Y. (2019). Making deep q-learning methods robust to time discretization. volume 97 of *Proceedings of Machine Learning Research*, pages 6096–6104, Long Beach, California, USA. PMLR.

Wang, H., Zariphopoulou, T., and Zhou, X. (2020). Reinforcement learning in continuous time and space: a stochastic control approach. *Journal of Machine Learning Research*, 21:1–34.

Yi, G. Y. (2017). *Statistical Analysis with Measurement Error or Misclassification: Strategy, Method and Application*. Springer.

Chenyang Jiang, Bowen Hu, and Yazhen Wang, University of Wisconsin-Madison

E-mail: cjiang77@wisc.edu, yzwang@stat.wisc.edu, bhu35@wisc.edu

Shang Wu, Fudan University

E-mail: shangwu@fudan.edu.cn