Fast Value Tracking for Deep Reinforcement Learning

Faming Liang

Purdue University

November 18, 2024

Reinforcement Learning

As a mathematical model, RL solves sequential decision-making problems by designing an agent that interacts with the environment, with the goal of learning an optimal policy that maximizes the expected total reward for the agent.

Prominent value-based algorithms:

- Temporal-difference (TD) learning (Sutton, 1988)
- State–action–reward–state–action (SARSA) (Sutton & Barto, 2018)
- Q-learning

Traditionally, these methods treat the state value (or Q-value) as a deterministic function, focusing on calculating point estimates of model parameters, thereby overlooking the inherent stochasticity in agent-environment interactions.

Reinforcement Learning

In the context of RL, a fair algorithm should exhibit the features:

(i) *Uncertainty quantification*: addressing the stochastic nature of the agent-environment interactions, thereby enhancing the robustness of the learned policy.

(ii) *Dynamicity*: addressing the dynamics of the agent-environment interaction system.

(iii) *Nonlinear approximation*: employing, for example, a deep neural network to approximate the value function.

(iv) *Computational efficiency*: scalable with respect to the model dimension and training sample size.

In RL, it is more suitable to treat values or model parameters as random variables rather than fixed unknowns, focusing on tracking dynamic changes rather than achieving point convergence during the policy learning process.

Kalman Temporal Difference (KTD)

KTD treats values or their parameters as random variables, focusing on the tracking property of the policy learning process. KTD conceptualizes RL as a state-space model:

$$\begin{aligned} \theta_t &= \theta_{t-1} + w_t, \\ \boldsymbol{r}_t &= h(\boldsymbol{x}_t, \theta_t) + \eta_t, \end{aligned} \tag{1}$$

where $\theta_t \in \mathbb{R}^p$ denotes the parameters at time step t with dimension p, $w_t \in \mathbb{R}^p$ and $\eta_t \in \mathbb{R}^n$ denote two independent multivariate Gaussian vectors, $\boldsymbol{x}_t$ denotes a set of states and actions collected at time step t, $\boldsymbol{r}_t \in \mathbb{R}^n$ denotes a vector of rewards, n denotes the number of samples, and $h(\cdot, \cdot)$ can be a linear or nonlinear function.

Kalman Temporal Difference (KTD)

- $h(\cdot)$ is linear: Kalman filter can be applied.
- $h(\cdot)$ is nonlinear: Unscented Kalman Filter (UKF), Extended Kalman Filter (EKF).

Both UKF and EKF can be computationally inefficient for high-dimensional parameter spaces.

Markov Decision Process

The RL procedure can be described by a Markov Decision Process represented by $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma\}$, where $\mathcal{S}$ is set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the state transition probability from state s to state s' by taking action a, denoted by $\mathcal{P}(s'|s, a)$, $r(s, a)$ is a random reward received from taking action a at state s, and $\gamma \in (0, 1)$ is a discount factor.

At each time step t, the agent observes state $s_t \in \mathcal{S}$ and takes action $a_t \in \mathcal{A}$ according to policy ρ with probability $P_\rho(a|s)$, then the environment returns a reward $r_t = r(s_t, a_t)$ and a new state $s_{t+1} \in \mathcal{S}$.

Markov Decision Process

For a given policy ρ, the performance is measured by the state value function $V_\rho(s) = \mathbb{E}_\rho[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s]$ and the state-action value function $Q_\rho(s, a) = \mathbb{E}_\rho[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a]$, which are called V-function and Q-function, respectively.

Both functions satisfy the Bellman equation:

$$\begin{aligned} V_\rho(s) &= \mathbb{E}_\rho[r(s, a) + \gamma V_\rho(s')], \\ Q_\rho(s, a) &= \mathbb{E}_\rho[r(s, a) + \gamma Q_\rho(s', a')], \end{aligned} \tag{2}$$

where $s' \sim \mathcal{P}(\cdot|s, a)$, $a \sim P_\rho(\cdot|s)$, $a' \sim P_\rho(\cdot|s')$, and the expectation $\mathbb{E}_\rho[\cdot]$ is taken over the transition probability distribution $\mathcal{P}$ for a given policy ρ.

Kalman Temporal Difference Algorithm

Let $\boldsymbol{s}_t = (s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(n)})^T$, $\boldsymbol{a}_t = (a_t^{(1)}, a_t^{(2)}, \ldots, a_t^{(n)})^T$, and $\boldsymbol{r}_t = (r_t^{(1)}, r_t^{(2)}, \ldots, r_t^{(n)})^T$ denote, respectively, a vector of n states, actions, rewards collected at time step t. Given the Bellman equation, the function $h(\boldsymbol{x}_t, \theta_t)$ in (1) can be expressed as

$$h(\boldsymbol{x}_t, \theta_t) = \begin{cases} V_{\theta_t}(\boldsymbol{s}_t) - \gamma V_{\theta_t}(\boldsymbol{s}_{t+1}), & \text{for } V\text{-function}, \\ Q_{\theta_t}(\boldsymbol{s}_t, \boldsymbol{a}_t) - \gamma Q_{\theta_t}(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}), & \text{for } Q\text{-function}, \end{cases} \tag{3}$$

where $\boldsymbol{x}_t = \{\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}\}$,
$V_{\theta_t}(\boldsymbol{s}_t) := (V_{\theta_t}(s_t^{(1)}), V_{\theta_t}(s_t^{(2)}), \ldots, V_{\theta_t}(s_t^{(n)}))^T$, and
$Q_{\theta_t}(\boldsymbol{s}_t, \boldsymbol{a}_t) := (Q_{\theta_t}(s_t^{(1)}, a_t^{(1)}), Q_{\theta_t}(s_t^{(2)}, a_t^{(2)}), \ldots, Q_{\theta_t}(s_t^{(n)}, a_t^{(n)}))^T$.

Kalman Temporal Difference Algorithm: Linearization

The fucntion $h(\boldsymbol{x}, \theta)$ is linearized based on the first-order Taylor expansion:

$$h(\boldsymbol{x}_t, \theta) \approx h(\boldsymbol{x}_t, \hat{\mu}_{t-1}) + \nabla_\theta h(\boldsymbol{x}_t, \hat{\mu}_{t-1})^T (\theta - \hat{\mu}_{t-1}),$$

where $\hat{\mu}_{t-1}$ denotes the estimator for the mean of θ_{t-1}.

Langevinized Kalman Temporal Difference (LKTD) algorithm

LKTD reformulates RL as the following state space model:

$$\begin{aligned} \theta_t &= \theta_{t-1} + \frac{\epsilon_t}{2} \nabla_\theta \log \pi(\theta_{t-1}) + w_t, \\ \boldsymbol{r}_t &= h(\boldsymbol{x}_t, \theta_t) + \eta_t, \end{aligned} \tag{4}$$

where $w_t \sim N(0, \epsilon_t I_p)$, $\pi(\theta)$ represents a prior density function we impose on θ, and $\{\epsilon_t : t = 1, 2, \dots\}$ is a positive sequence decaying to zero.

LKTD integrates the KTD and LEnKF algorithm, leading to an effective algorithm for RL.

LKTD Algorithm

By the state augmentation approach, we define

$$\varphi_t = \begin{pmatrix} \theta_t \\ \xi_t \end{pmatrix}, \quad \xi_t = h(\boldsymbol{x}_t; \theta_t) + u_t, \quad u_t \sim N(0, \alpha\sigma^2 I_n), \qquad (5)$$

where ξ_t is an n-dimensional vector, and $0 < \alpha < 1$ is a pre-specified constant.

Based on Langevin dynamics, we can reformulate (4) as

$$\begin{aligned} \varphi_t &= \varphi_{t-1} + \frac{\epsilon_t}{2} \frac{n}{\mathcal{N}} \nabla_\varphi \log \pi(\varphi_{t-1}) + \tilde{w}_t, \\ \boldsymbol{r}_t &= H_t \varphi_t + v_t, \end{aligned} \qquad (6)$$

where $\mathcal{N} > 0$, $\tilde{w}_t \sim N(0, \frac{n}{\mathcal{N}} B_t)$, $B_t = \epsilon_t I_{\tilde{p}}$, $\tilde{p} = p + n$ is the dimension of φ_t; $H_t = (\mathbf{0}, I_n)$ such that $H_t \varphi_t = \xi_t$; $v_t \sim N(0, (1-\alpha)\sigma^2 I_n)$, which is independent of $\tilde{w}_t$ for all t. We call $\mathcal{N}$ the pseudo-population size, which scales uncertainty of the estimator of the system.

LKTD Algorithm

- Initialization: Draw $\theta_0^a \in \mathbb{R}^p$ drawn from the prior distribution $\pi(\theta)$.

  Do the following for $t = 1, 2, \ldots, T$:

  - Sampling: With policy $\rho_{\theta_{t-1}^a}$, generate a set of n transition tuples, denoted by $\boldsymbol{z}_t = (\boldsymbol{r}_t, \boldsymbol{x}_t) := \{r_t^{(j)}, x_t^{(j)}\}_{j=1}^n$.

    Do the following for $k = 1, 2, \ldots, \mathcal{K}$:

    - Presetting: Set $B_{t,k} = \epsilon_{t,k} I_{\tilde{p}}$, $R_t = 2(1-\alpha)\sigma^2 I$, and the Kalman gain matrix $K_{t,k} = B_{t,k} H_t^\top (H_t B_{t,k} H_t^\top + R_t)^{-1}$.
    - Forecast: Draw $\tilde{w}_{t,k} \sim N_p(0, \frac{n}{\mathcal{N}} B_{t,k})$ and calculate

      $$\varphi_{t,k}^f = \varphi_{t,k-1}^a + \frac{\epsilon_{t,k}}{2} \frac{n}{\mathcal{N}} \nabla_\varphi \log \pi(\varphi_{t,k-1}^a) + \tilde{w}_{t,k}, \tag{7}$$

      where $\varphi_{t,0}^a = ({\theta_{t-1,\mathcal{K}}^a}^\top, \boldsymbol{r}_t^\top)^\top$ if $k = 1$.
    - Analysis: Draw $v_{t,k} \sim N_n(0, \frac{n}{\mathcal{N}} R_t)$ and calculate

      $$\varphi_{t,k}^a = \varphi_{t,k}^f + K_{t,k}(\boldsymbol{r}_t - H_t \varphi_{t,k}^f - v_{t,k}) = \varphi_{t,k}^f + K_{t,k}(\boldsymbol{r}_t - \boldsymbol{r}_{t,k}^f). \tag{8}$$

Convergence theory

Lemma 1

The LKTD algorithm implements a preconditioned SGLD algorithm, for which

$$\varphi_t^a = \varphi_{t-1}^a + \frac{\epsilon_t}{2}\Sigma_t \nabla_\varphi \log \pi(\varphi_{t-1}^a | \boldsymbol{z}_t) + e_t, \tag{9}$$

where $\boldsymbol{z}_t = (\boldsymbol{r}_t, \boldsymbol{x}_t)$, $\Sigma_t = \frac{n}{\mathcal{N}}(I - K_t H_t)$ *is a constant matrix given* φ_t, $e_t \sim N(0, \epsilon_t \Sigma_t)$, *and the gradient term* $\nabla_\varphi \log \pi(\varphi_{t-1}^a | \boldsymbol{z}_t)$ *is given by* $\nabla_\varphi \log \pi(\varphi_{t-1}^a | \boldsymbol{z}_t) = \frac{\mathcal{N}}{n} \sum_{i=1}^n \nabla_\varphi \log \pi(z_t^{(i)} | \varphi_{t-1}^a) + \nabla_\varphi \log \pi(\varphi_{t-1}^a)$.

Convergence theory: On-policy setting

Theorem 2

Consider a SGLD sampler with a polynomailly-decay learning rate $\epsilon_k = \frac{\epsilon_0}{k^{\varpi}}$ *for some* $\varpi \in (0,1)$. *Suppose that the environment is stationary and regularity conditions hold. Then there exist constants* (C_0, C_1, C_2, C_3) *such that for all* $K \in \mathbb{N}$, *the 2-Wasserstein distance between* μ_K *and* $\nu_{\mathcal{N}}$ *can be bounded by*

$$\begin{aligned} \mathcal{W}_2(\mu_K, \nu_{\mathcal{N}}) \leq (12 + C_2\epsilon_0(\frac{1}{1-\varpi}K^{1-\varpi}))^{\frac{1}{2}} \cdot [(C_1\epsilon_0^2(\frac{2\varpi}{2\varpi-1}) + \delta C_0(\frac{\epsilon_0}{1-\varpi}K^{1-\varpi}))^{\frac{1}{2}} \\ + (C_1\epsilon_0^2(\frac{2\varpi}{2\varpi-1}) + \delta C_0(\frac{\epsilon_0}{1-\varpi}K^{1-\varpi}))^{\frac{1}{4}}] + C_3 \exp(-\frac{1}{\beta c_{LS}}(\frac{\epsilon_0}{1-\varpi}K^{1-\varpi}) \end{aligned} \tag{10}$$

where $\mu_K(\theta)$ *denotes the probability law of* θ_K, $\nu_{\mathcal{N}}(\theta) \propto \exp(-\beta\mathcal{G}(\theta))$, $\mathcal{G}(\theta) = O(\mathcal{N})$ *is the anti-derivative of* $g(\theta)$, *i.e.,* $\nabla_\theta \mathcal{G}(\theta) = g(\theta)$, *and* c_{LS} *denotes a logarithmic Sobolev constant satisfied by the* $\nu_{\mathcal{N}}$.

Convergence theory: with replay buffer

Theorem 3

Let $\{\theta_t\}_{t=1}^T$ be a sequence of updates generated from LKTD with replay buffer. At each time t, the transition tuple z_t is sampled from the replay buffer $\bar{\pi}(z_t|\boldsymbol{\theta}_{t-1}^R)$. In addition to the assumptions in theorem 2, we further assume the following holds:

1. *(Lipschitz)* $\int_{\mathcal{Z}} |\pi(z|\theta) - \pi(z|\vartheta)|^2 dz \leq L\|\theta - \vartheta\|^2$;
2. *(Integrability)* $\int_{\mathcal{Z}} \|G(\theta, z)\|^2 dz \leq M$ *and* $\int_{\mathcal{Z}} \|G(\theta, z)\|^2 \pi(z|\theta) dz \leq M$, $\forall \theta \in \Theta$.

Then for a bounded test function ϕ, the bias of the LKTD can be bounded as:

$$|\mathbb{E}\hat{\phi} - \bar{\phi}| = O(\frac{1}{S_T} + \frac{\sum_{t=1}^T \epsilon_t^2}{S_T}), \quad \mathbb{E}(\hat{\phi} - \bar{\phi})^2 = O(\frac{1}{S_T} + \frac{\sum_{t=1}^T \epsilon_t^2}{S_T^2} + \frac{(\sum_{t=1}^T \epsilon_t^2)^2}{S_T^2}) \tag{11}$$

Indoor Escape Environment

For this environment, the state space consists of 100 grids and the agent's objective is to navigate to the goal positioned at the top right corner.

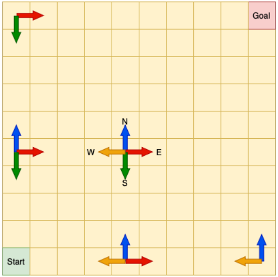

Figure 1: Indoor escape environment

Indoor Escape Environment

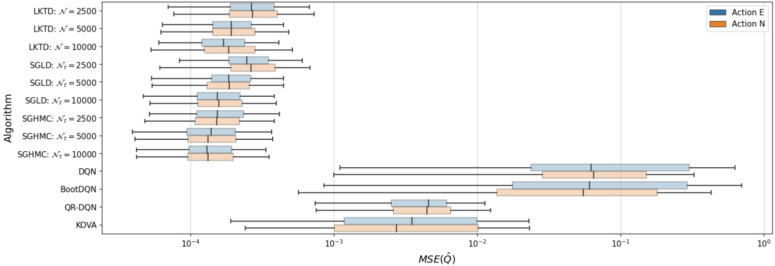

Figure 2: Boxplots for $\text{MSE}(\hat{Q}_a)$ (for $a \in \{N, E\}$)

Indoor Escape Environment

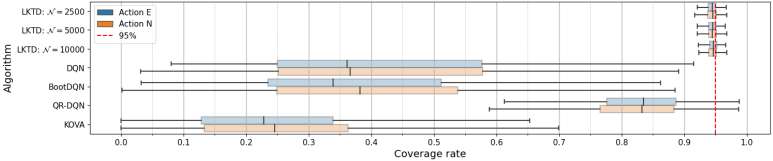

Figure 3: Boxplots for coverage rates (for $a \in \{N, E\}$))

Indoor Escape Environment

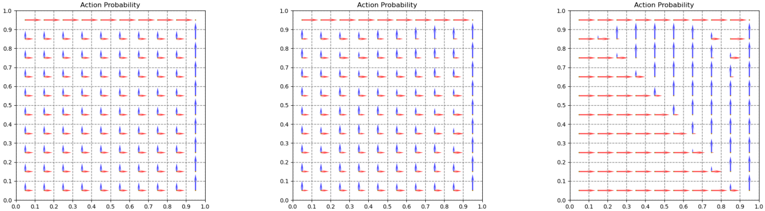

Figure 4: Mean policy probabilities for the indoor escape environment: (a) known optimal solution; (b) learned by LKTD; (c) learned by DQN, failing to explore different policies.

Classical Control Problems

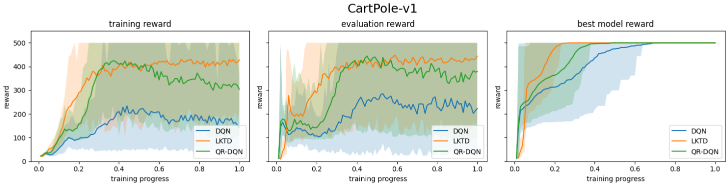

Figure 5: CartPole-v1: The left plot shows the cumulative rewards obtained during the training process, the middle plot shows the testing performance without random exploration, and the right plot shows the performance of best model learned up to the time t.

Conclusion

- By redefining the state-space model, we enable SGMCMC algorithms, such as LKTD and SGLD, to converge to the accurate posterior distribution under mild conditions.
- Our LKTD algorithm demonstrates greater computational efficiency and circumvents potential matrix degeneration issues by eliminating the need for linearization.
- Compared to DQN and its variants, our framework not only provides more precise point estimates of Q-values but also generates accurate prediction intervals for value tracking.

References

Frank Shih and Faming Liang (2024). Fast Value Tracking for Deep Reinforcement Learning. *ICLR 2024*.