Reinforcement Learning 3

Kintan Saha

January 10, 2026

Outline

TD Learning: Optimization View

Goal: Evaluate a fixed policy μ by learning its value function V^μ from samples.

We begin with the idealized objective

$$f(x) \;:=\; \frac{1}{2}\sum_{s\in\mathcal{S}} d(s)\big(V^\mu(s) - x(s)\big)^2 = \frac{1}{2}\,\|V^\mu - x\|_d^2\,,$$

where d is a weighting distribution on $\mathcal{S}$ (e.g. stationary distribution under μ), and $\|y\|_d^2 := \sum_s d(s)y(s)^2$.

Observation: The unique minimizer of f is $x = V^\mu$.

From Gradient Descent to TD

The gradient of f (tabular) is

$$\nabla f(x)(s) = d(s)\big(x(s) - V^{\mu}(s)\big).$$

A formal gradient descent update is

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n).$$

Issue: V^{μ} is unknown.

$$x_{n+1} = x_n + \alpha_n \sum_{s,a,s'} d(s)\mu(a|s)P(s'|s,a)\Big[r(s,a) + \gamma V^{\mu}(s') - V^{\mu}(s)\Big] e_s$$

Use the Bellman evaluation equation:

$$V^{\mu}(s) = \sum_{a\in\mathcal{A}} \mu(a|s)\Big(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V^{\mu}(s')\Big).$$

Substitute this into the update, then **bootstrap** by replacing $V^{\mu}(s')$ with the current estimate $x_n(s')$.

Replacing Expectations with Samples: TD(0)

$$x_{n+1} = x_n + \alpha_n \sum_{s,a,s'} d(s)\mu(a|s)P(s'|s,a)\left[r(s,a) + \gamma x_n(s') - x_n(s)\right] e_s$$

After bootstrapping, the (still expectation-based) drift becomes implementable via samples from the stationary one-step distribution under μ:

$$S_n \sim d, \quad A_n \sim \mu(\cdot|S_n), \quad S_{n+1} \sim P(\cdot|S_n, A_n), \quad R_n = r(S_n, A_n).$$

Replacing expectations by a single sample yields the TD(0) update:

$$x_{n+1}(S_n) = x_n(S_n) + \alpha_n\Big(R_n + \gamma x_n(S_{n+1}) - x_n(S_n)\Big),$$

and $x_{n+1}(s) = x_n(s)$ for all $s \neq S_n$.

Remark: TD is a stochastic recursion (not true GD on the original f).

i.i.d sampling of (S_n, A_n, S_{n+1})

TD Learning as a Fixed-Point Scheme

The value function V^μ is the unique fixed point of the Bellman evaluation operator:

$$T_\mu V^\mu = V^\mu,$$

where

$$(T_\mu V)(s) = \sum_a \mu(a|s)\Big(r(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s')\Big).$$

To solve the fixed-point equation $T_\mu x = x$, a natural approach is the *fixed-point iteration*

$$x_{n+1} = x_n + a_n\big(T_\mu x_n - x_n\big),$$

where $\{a_n\}$ is a stepsize sequence.

$x_{n+1} = x_n$ $\quad T_\mu x_n = x_n \Rightarrow x_n = V_\mu$

This iterative procedure will terminate at the fixed point x when $T_\mu x = x$.

From Fixed-Point Iteration to TD(0)

Substituting the definition of T_μ into the fixed-point iteration gives

$$x_{n+1}(s) = x_n(s) + a_n\Big(r_\mu(s) + \gamma \sum_{s'} P_\mu(s'|s)x_n(s') - x_n(s)\Big),$$

where

$$r_\mu(s) := \sum_a \mu(a|s)r(s,a), \qquad P_\mu(s'|s) := \sum_a \mu(a|s)P(s'|s,a).$$

This update still involves expectations w.r.t. P_μ, which are unknown.

Replacing these expectations by a single sample (S_n, A_n, R_n, S_{n+1}) drawn under policy μ yields

$$x_{n+1}(S_n) = x_n(S_n) + a_n\Big(R_n + \gamma x_n(S_{n+1}) - x_n(S_n)\Big),$$

with $x_{n+1}(s) = x_n(s)$ for all $s \neq S_n$.

This is precisely the TD(0) update.

Stochastic Approximation

Many noisy iterative algorithms (including TD) are instances of *stochastic approximation (SA)*:

$$x_{n+1} = x_n + a(n)\big(h(x_n) + M_{n+1}\big), \qquad x_n \in \mathbb{R}^d,$$

where:

- $a(n) > 0$ is the stepsize,
- $h : \mathbb{R}^d \to \mathbb{R}^d$ is the mean drift,
- $\{M_n\}$ is a noise sequence with $\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] = 0$.

ODE Method

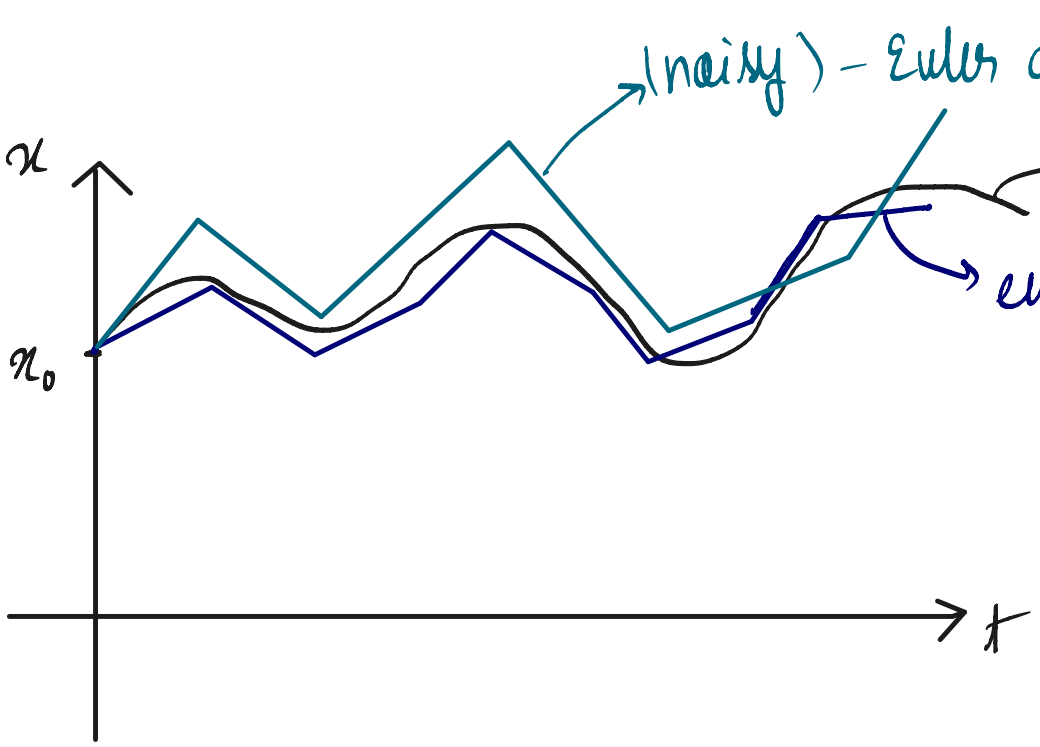
$$\dot{x} = h(x) \Rightarrow \lim_{\delta \to 0} \frac{x(t+\delta) - x(t)}{\delta} = h(x(t))$$

$$\Rightarrow x(t+\delta) \approx x(t) + \delta h(x(t))$$

Consider an ODE $\dot{x} = h(x)$

The Euler discretisation of the ODE : $x_{t+1} = x_t + \alpha_t h(x_t)$

(Noisy)-Euler discretisation of the ODE : $x_{t+1} = x_t + \alpha_t [h(x_t) + \text{"noise"}]$



(Q2) Does the "noisy" Euler discretisation become "arbitarily close" to the ODE ?

(Q1) Does the Euler discretisation become "arbitarily close" to the ODE trajectory ?

Borkar–Meyn Theorem (Result)

Consider the SA recursion

$$x_{n+1} = x_n + a(n)\big(h(x_n) + M_{n+1}\big).$$

(Borkar–Meyn / ODE method). Under assumptions A1–A4, the interpolated trajectory tracks the ODE

$$\dot{x}(t) = h(x(t)).$$

If the ODE has a unique globally asymptotically stable equilibrium $x^\star$, then

$$x_n \xrightarrow{a.s.} x^\star.$$

Assumptions (A1–A3)

(A1) Step sizes (Robbins–Monro):

$$a(n) > 0, \qquad \sum_{n=0}^{\infty} a(n) = \infty, \qquad \sum_{n=0}^{\infty} a(n)^2 < \infty.$$

(A2) Drift regularity (Lipschitz):

$$\|h(x) - h(y)\| \le L\,\|x - y\| \qquad \forall x, y \in \mathbb{R}^d.$$

(A3) Martingale-difference noise + bounded conditional 2nd moment:

$$\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] = 0, \qquad \mathbb{E}[\|M_{n+1}\|^2 \mid \mathcal{F}_n] \le C(1 + \|x_n\|^2).$$

Remark: Choice of Norm

Throughout the assumptions above, the specific choice of norm $\|\cdot\|$ is immaterial.

Reason Since the iterates $x_n \in \mathbb{R}^d$ evolve in a finite-dimensional space, all norms on $\mathbb{R}^d$ are equivalent. That is, for any two norms $\|\cdot\|_a$ and $\|\cdot\|_b$, there exist constants $c_1, c_2 > 0$ such that

$$c_1\|x\|_a \le \|x\|_b \le c_2\|x\|_a \qquad \forall x \in \mathbb{R}^d.$$

$\|x\|_a \le \|y\|_a \iff \|x\|_b \le \|y\|_b$

We will work with the ℓ_∞ norm

$\|x\|_\infty = \max_{i \in [d]} |x_i|$

Consequences

- Boundedness of $\{x_n\}$ in one norm implies boundedness in any other norm.
- Lipschitz continuity of h in one norm implies Lipschitz continuity in any other norm (with possibly different constants).
- Almost sure convergence and stability properties are invariant to the choice of norm.

Hence, in verifying Assumptions A1–A3, we may work with any convenient norm (e.g. ℓ_∞ or ℓ_2) without loss of generality.

Assumptions (A4 / A5): Stability and Scaled ODE

(A4) Stability of iterates:

$$\sup_{n \geq 0} \|x_n\| < \infty \quad \text{almost surely.}$$

(A5) Scaled drift / ODE at infinity: Assume the limit exists uniformly on compacts,

$$h_\infty(x) := \lim_{c \to \infty} \frac{h(cx)}{c},$$

and the ODE

$$\dot{x} = h_\infty(x)$$

has the origin as a *globally asymptotically stable equilibrium (GASE)*.

Borkar–Meyn Theorem Under assumptions A1–A4, or A1–A3+A5:

- the interpolated trajectory tracks the ODE $\dot{x} = h(x)$.

$$x_{n+1} = x_n + a(n)\left[h(x_n) + M_{n+1}\right]$$

$$\|x_{n+1}\| \leq \|x_n\| + a(n)\left[\|h(x_n)\| + \|M_{n+1}\|\right] \quad \rightarrow c(1+\|x_n\|)$$

$$\Leftrightarrow \|x_{n+1}\|^2 \leq 2\left[\|x_n\|^2 + 2a(n)^2\left[\frac{\|h(x_n)\|^2}{+\|M_{n+1}\|^2}\right]\right]$$

$$(a+b)^2 \leq 2(a^2+b^2)$$

① $n \to \infty$ x_n are close to the trajectory

② $\{x_n\}$ will converge to ——— set

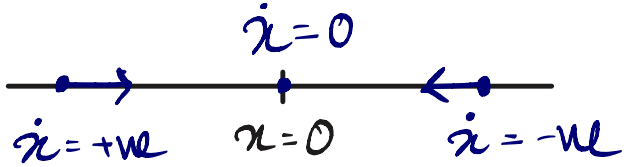
of $\dot{x} = h(x)$ ——— eq^m points of the ODE

$h(x) = ax + b \quad \in \mathbb{R}^1$

$h_\infty(x) = \lim_{c \to \infty} \frac{acx + b}{c} = ax \qquad h_\infty(x) = ax$

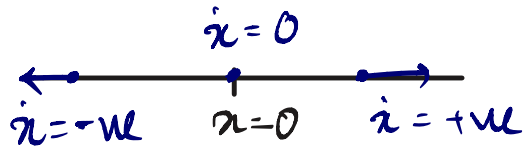
Primer on ODEs

Consider ① $\dot{x} = -x \Rightarrow x(t) = x(0)\, e^{-t}$ → trajectory of the ODE



→ slight perturbation around eq^m pt results in trajectory returning to 0 ⇒ stable equilibrium

② $\dot{x} = x$



→ slight perturbation around eq^m point results in trajectory not returning to 0

⇒ unstable equilibrium

Globally asymptotically stable equilibrium (GASE)

↓

starting from any where on $\mathbb{R}$, the trajectory of the ODE goes to the eq^m point → 0 is GASE for ① not ②

Q. Is there a way to formalise "start anywhere, terminate in the eqm point"?

Consider $\dot{x} = -x$

Define $V(x) = \frac{1}{2}x^2 \Rightarrow \frac{dV}{dt} = \frac{dV}{dx}\cdot\frac{dx}{dt} = x\cdot\dot{x} = -x^2$

$\Rightarrow \dot{V} = -x^2 \leq 0 \quad \forall x$

$V \geq 0 \quad \forall x$

V is a "Lyapunov function" $\equiv$ potential energy

This potential energy is minimised at 0 and its' time derivative is negative is negative everywhere other than 0

$\Rightarrow$ starting from anywhere, the potential energy decreases (since $\dot{V} < 0$) until it reaches 0 (at which $\dot{V} = 0$)

Theorem: If such a V exists and $\{x \mid \dot{V}(x) = 0\}$ is a singleton then $\{x \mid \dot{V}(x) = 0\}$ is a GASE for $\dot{x} = h(x)$

Q. For $\dot{x} = x$, why does $V = \frac{-1}{2}x^2$ not work?

Answer: ??

Outline

TD Learning as SA (Vector Form)

Let $d := |\mathcal{S}|$ and identify $x \in \mathbb{R}^d$ with the tabular vector $(x(s))_{s\in\mathcal{S}}$. Let $e_s \in \mathbb{R}^d$ be the standard basis vector.

TD(0) update:

$$x_{n+1} = x_n + a(n)\, e_{S_n}\Big(\overbrace{R_n + \gamma x_n(S_{n+1}) - x_n(S_n)}^{\chi}\Big).$$

$h = \mathbb{E}\chi$

$M = \chi - \mathbb{E}\chi$

$\mathbb{E}M = 0$

We will write this as

$$x_{n+1} = x_n + a(n)\big(h(x_n) + M_{n+1}\big),$$

and verify A1–A4.

SA-I: Drift as Conditional Expectation; Noise as Centering

Define the one-step random update using the sample $\xi_{n+1} := (S_n, A_n, R_n, S_{n+1})$:

$$H(x, \xi_{n+1}) := e_{S_n}\Big(R_n + \gamma x(S_{n+1}) - x(S_n)\Big) \in \mathbb{R}^d.$$

Let $\{\mathcal{F}_n\}$ be the natural filtration generated by the algorithm:

$$\mathcal{F}_n := \sigma\big(x_0, \xi_1, \ldots, \xi_n\big).$$

Define the drift as the *conditional expectation*

$$h(x_n) := \mathbb{E}[\, H(x_n, \xi_{n+1}) \,|\, \mathcal{F}_n],$$

and define the noise by

$$M_{n+1} := H(x_n, \xi_{n+1}) - h(x_n).$$

With these definitions, the TD recursion can be written exactly as

$$x_{n+1} = x_n + a(n)\big(h(x_n) + M_{n+1}\big).$$

SA-II: Explicit Form of the Drift $h(x)$

Define the policy-induced reward and transition kernel:

$$r_\mu(s) := \sum_{a \in \mathcal{A}} \mu(a|s)\, r(s,a), \qquad P_\mu(s'|s) := \sum_{a \in \mathcal{A}} \mu(a|s)\, P(s'|s,a).$$

$h(x)(s) = \mathbb{E}\left[R_n + \gamma x(S_{n+1}) - x(S_n) \mid \mathcal{F}_n\right]$

Then the Bellman evaluation operator is

$$(T_\mu x)(s) = r_\mu(s) + \gamma \sum_{s'} P_\mu(s'|s) x(s').$$

Under the stationary sampling distribution $S_n \sim d$,

$$h(x)(s) = d(s)\big((T_\mu x)(s) - x(s)\big).$$

We now verify A1–A4.

Verification – A2 (Step Sizes)

Assume the stepsizes $\{a(n)\}$ satisfy the Robbins–Monro conditions:

$$a(n) > 0, \qquad \sum_{n=0}^{\infty} a(n) = \infty, \qquad \sum_{n=0}^{\infty} a(n)^2 < \infty.$$

Hence A2 holds.

Verification – A2 (Lipschitz Drift: Pointwise Bound)

Recall

$$h(x)(s) = d(s)\Big(r_\mu(s) + \gamma \sum_{s'} P_\mu(s'|s)x(s') - x(s)\Big).$$

For any $x, y \in \mathbb{R}^d$, the difference satisfies

$$h(x)(s) - h(y)(s) = d(s)\Big(\gamma \sum_{s'} P_\mu(s'|s)\big(x(s') - y(s')\big) - (x(s) - y(s))\Big).$$

Taking absolute values and using the triangle inequality,

$$|h(x)(s) - h(y)(s)| \le d(s)\Big(\gamma \sum_{s'} P_\mu(s'|s)\,|x(s') - y(s')| + |x(s) - y(s)|\Big).$$

Since $P_\mu(\cdot|s)$ is a probability distribution, $\sum_{s'} P_\mu(s'|s) \le 1$, $|x(s') - y(s')| \le \|x-y\|_\infty$, $|x(s) - y(s)| \le \|x-y\|_\infty$

$$\sum_{s'} P_\mu(s'|s)\,|x(s') - y(s')| \le \|x - y\|_\infty.$$

Therefore, for all $s \in \mathcal{S}$,

$$|h(x)(s) - h(y)(s)| \le d(s)(\gamma + 1)\,\|x - y\|_\infty.$$

Verification – A2 (Lipschitz Drift: Norm Bound)

From the pointwise bound,

$$|h(x)(s) - h(y)(s)| \leq d(s)(\gamma + 1)\,\|x - y\|_\infty \quad \forall s \in \mathcal{S}.$$

Taking the ℓ_∞-norm over s,

$$\|h(x) - h(y)\|_\infty = \max_{s \in \mathcal{S}} |h(x)(s) - h(y)(s)| \leq (\gamma + 1)\,\|x - y\|_\infty \max_s d(s).$$

Since d is a probability distribution on $\mathcal{S}$,

$$\max_s d(s) \leq 1.$$

Hence,

$$\|h(x) - h(y)\|_\infty \leq (\gamma + 1)\,\|x - y\|_\infty.$$

Conclusion: h is globally Lipschitz with constant $L = \gamma + 1$.

Verification – A3 (I): Filtration and Conditional Mean-Zero

Define the natural filtration (information up to time n)

$$\mathcal{F}_n := \sigma\big(x_0, \xi_1, \ldots, \xi_n\big) = \sigma\big(x_0, (S_0, A_0, R_0, S_1), \ldots, (S_{n-1}, A_{n-1}, R_{n-1}, S_n)\big).$$

By construction, x_n is $\mathcal{F}_n$-measurable.

Recall

$$M_{n+1} = H(x_n, \xi_{n+1}) - h(x_n), \qquad h(x_n) = \mathbb{E}[H(x_n, \xi_{n+1}) \mid \mathcal{F}_n].$$

Therefore,

$$\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] = \mathbb{E}[H(x_n, \xi_{n+1}) \mid \mathcal{F}_n] - h(x_n) = 0.$$

So $\{M_n\}$ is a martingale-difference noise sequence.

Verification – A3 (II): Conditional Second Moment Setup

We show a bound of the form

$$\mathbb{E}[\|M_{n+1}\|^2 \mid \mathcal{F}_n] \leq C(1 + \|x_n\|^2).$$

Using $\|u - v\|^2 \leq 2\|u\|^2 + 2\|v\|^2$,

$$\|M_{n+1}\|^2 = \|H(x_n, \xi_{n+1}) - h(x_n)\|^2 \leq 2\|H(x_n, \xi_{n+1})\|^2 + 2\|h(x_n)\|^2 .$$

Thus it suffices to bound $\mathbb{E}[\|H(x_n, \xi_{n+1})\|^2 \mid \mathcal{F}_n]$ and $\|h(x_n)\|^2$ by $O(1 + \|x_n\|^2)$.

Assume bounded rewards: $|r(s, a)| \leq R_{\max}$, hence $|R_n| \leq R_{\max}$ a.s.

Verification – A3 (III): Bounding $\mathbb{E}[\|H(x_n, \xi_{n+1})\|^2 \mid \mathcal{F}_n]$

Recall

$$H(x_n, \xi_{n+1}) = e_{S_n}\Big(R_n + \gamma x_n(S_{n+1}) - x_n(S_n)\Big).$$

Since $\|e_{S_n}\| = 1$, Denoting $\|\cdot\|_\infty$ as $\|\cdot\|$

$$\|H(x_n, \xi_{n+1})\|^2 = \Big(R_n + \gamma x_n(S_{n+1}) - x_n(S_n)\Big)^2.$$

Using $(a+b+c)^2 \le 3(a^2+b^2+c^2)$ and $|R_n| \le R_{\max}$,

$$\|H(x_n, \xi_{n+1})\|^2 \le 3R_{\max}^2 + 3\gamma^2 x_n(S_{n+1})^2 + 3x_n(S_n)^2.$$

Since $x_n(S_n)^2 \le \|x_n\|^2$ and $x_n(S_{n+1})^2 \le \|x_n\|^2$,

$$\|H(x_n, \xi_{n+1})\|^2 \le 3R_{\max}^2 + 3(\gamma^2+1)\|x_n\|^2.$$

Taking conditional expectation,

$$\mathbb{E}\big[\|H(x_n, \xi_{n+1})\|^2 \mid \mathcal{F}_n\big] \le 3R_{\max}^2 + 3(\gamma^2+1)\|x_n\|^2.$$

Verification – A3 (IV): Bounding $\|h(x_n)\|^2$

Recall that the drift is

$$h(x_n) = \mathbb{E}[H(x_n, \xi_{n+1}) \mid \mathcal{F}_n] .$$

By Jensen's inequality,

$$\|h(x_n)\|^2 = \|\mathbb{E}[H(x_n, \xi_{n+1}) \mid \mathcal{F}_n]\|^2 \leq \mathbb{E}\left[\|H(x_n, \xi_{n+1})\|^2 \mid \mathcal{F}_n\right] .$$

Using the bound from the previous slide,

$$\|h(x_n)\|^2 \leq 3R_{\max}^2 + 3(\gamma^2 + 1)\|x_n\|^2.$$

Hence there exists a constant $C_1 > 0$ such that

$$\|h(x_n)\|^2 \leq C_1\left(1 + \|x_n\|^2\right).$$

Verification – A4 (Scaled Drift and Stability)

Recall the drift

$$h(x)(s) = d(s)\Big(r_\mu(s) + \gamma(P_\mu x)(s) - x(s)\Big).$$

Define the scaled drift

$$h_\infty(x) := \lim_{c\to\infty} \frac{h(cx)}{c}.$$

Since the reward term is $O(1)$, it vanishes under scaling, and we obtain

$$h_\infty(x)(s) = d(s)\Big(\gamma(P_\mu x)(s) - x(s)\Big).$$

In vector form,

$$\dot{x} = h_\infty(x) = D(\gamma P_\mu - I)x, \qquad D := \mathrm{diag}(d).$$

This is a linear ODE. Under the discounted setting $\gamma \in (0,1)$, all eigenvalues of $\gamma P_\mu - I$ have strictly negative real parts. Hence, the origin is a *globally asymptotically stable equilibrium (GASE)* of the scaled ODE.

Verification – A4: Implication and Conclusion

We know from A5 if the scaled ODE has the origin as a GASE, then the stochastic approximation iterates $\{x_n\}$ are *stable*, i.e.

$$\sup_{n \geq 0} \|x_n\| < \infty \quad \text{a.s.}$$

Therefore, Assumption A4 is satisfied.

Conclusion (Borkar–Meyn). Since Assumptions A1–A4 hold, the TD iterates track the ODE

$$\dot{x} = h(x)$$

and converge almost surely to its equilibrium point $x^\star$.

Finally, $h(x) = 0$ implies for all $s \in \mathcal{S}$,

$$0 = d(s)\big((T_\mu x)(s) - x(s)\big) \quad \Rightarrow \quad T_\mu x = x \quad \Rightarrow \quad x = V^\mu.$$

Thus TD(0) converges almost surely to V^μ.

Remarks: Beyond One-Step TD

So far, we have seen two extremes:

- **Monte Carlo methods:** use the full infinite-horizon return

$$\sum_{t=0}^{\infty} \gamma^t R_t,$$

- **TD(0):** use a single transition (S_n, A_n, R_n, S_{n+1}) and bootstrap from $x_n(S_{n+1})$.

This raises a natural question:

Can we use **finite-horizon transition data** *to interpolate between these extremes?*

Concretely, instead of a single transition, one may use k-step data of the form

$$(S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_k),$$

and form a k-step return

$$R_1 + \gamma R_2 + \cdots + \gamma^{k-1} R_k + \gamma^k x_n(S_k).$$

TD(λ) provides a principled way to combine such multi-step returns by exponentially weighting different horizons.

Remarks: Why Care About Multi-Step Methods?

One might ask: *If TD(0) converges to V^μ asymptotically, why do we need multi-step methods at all?*

Answer: asymptotic convergence alone is often insufficient.

In practice:

- Algorithms are run for *finite time*, often under strict sample budgets.
- Performance depends on the *speed of convergence*, not just the limit.
- One-step TD can converge slowly due to bias from excessive bootstrapping.

Multi-step methods trade off bias and variance:

- Longer returns reduce bias but increase variance.
- Shorter returns reduce variance but increase bias.

TD(λ) allows us to control this trade-off and often yields better *finite-time performance*, even though all methods share the same asymptotic limit V^μ.

Key takeaway: asymptotic convergence guarantees correctness, but finite-time behavior determines practical effectiveness.

Outline

Why Function Approximation?

Recall the TD(0) update for policy evaluation:

$$x_{n+1}(S_n) = x_n(S_n) + a_n\big(R_n + \gamma x_n(S_{n+1}) - x_n(S_n)\big),$$

with $x_n(s)$ maintained for *every* state $s \in \mathcal{S}$.

This tabular representation requires:

- storing one parameter per state,
- updating values state-by-state.

Problem: When the state space is large (or infinite), this becomes infeasible.

Example:

- Robot control with continuous position and velocity.
- Navigation problems with continuous state variables.
- Games or MDPs with combinatorial state descriptions.

Function Approximation

To handle large or continuous state spaces, we approximate the value function V^μ using a restricted function class.

Instead of learning $V^\mu(s)$ for each s, we learn

$$V^\mu(s) \approx \hat{V}_\theta(s),$$

where $\hat{V}_\theta$ is parameterized by θ.

A good function class should be:

- low-dimensional,
- smooth / regular,
- amenable to optimization and analysis.

Two common choices:

- **Linear function approximation**,
- **Neural networks** (nonlinear approximation).

Linear Function Approximation

In linear function approximation, we represent the value function as

$$\hat{V}_\theta(s) = \phi(s)^\top \theta,$$

where:

- $\phi(s) \in \mathbb{R}^d$ is a feature vector,
- $\theta \in \mathbb{R}^d$ is the parameter vector.

Let $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$ be the feature matrix whose s-th row is $\phi(s)^\top$.

The set of representable value functions is

$$\mathcal{V}_\Phi := \{ \Phi\theta : \theta \in \mathbb{R}^d \},$$

a d-dimensional subspace of $\mathbb{R}^{|\mathcal{S}|}$.

Geometry of Linear Policy Evaluation

Recall the objective for policy evaluation:

$$\min_{x \in \mathbb{R}^{|\mathcal{S}|}} \frac{1}{2} \| V^\mu - x \|_d^2.$$

Under linear function approximation, we restrict $x = \Phi\theta$, giving

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2} \| V^\mu - \Phi\theta \|_d^2.$$

Geometric interpretation:

- $\mathcal{V}_\Phi$ is a linear subspace,
- $\Phi\theta^\star$ is the closest point in $\mathcal{V}_\Phi$ to V^μ,
- equivalently, $\Phi\theta^\star$ is the orthogonal projection of V^μ onto $\mathcal{V}_\Phi$ (with respect to the d-weighted inner product).

Projected Bellman Equation

In general, $V^\mu \notin \mathcal{V}_\Phi$, so we cannot solve

$$T_\mu x = x$$

within the function class.

Instead, we seek $\hat{V} = \Phi\theta$ satisfying the *projected Bellman equation*:

$$\Phi\theta = \Pi\big(T_\mu(\Phi\theta)\big),$$

where Π denotes projection onto $\mathcal{V}_\Phi$ (with respect to the d-weighted norm).

This equation characterizes the best approximation to V^μ within the linear function class.

TD Learning under Linear Function Approximation

With $\hat{V}_\theta(s) = \phi(s)^\top \theta$, the TD(0) update becomes

$$\theta_{n+1} = \theta_n + a_n \, \phi(S_n) \Big(R_n + \gamma \phi(S_{n+1})^\top \theta_n - \phi(S_n)^\top \theta_n \Big).$$

This update is obtained by:

- performing gradient descent in parameter space,
- bootstrapping the Bellman target,
- replacing expectations by samples, as in the tabular case.

The recursion now evolves in $\mathbb{R}^d$ instead of $\mathbb{R}^{|\mathcal{S}|}$.

Convergence of Linear TD Learning

The linear TD(0) update admits a stochastic approximation form:

$$\theta_{n+1} = \theta_n + a_n\big(h(\theta_n) + M_{n+1}\big),$$

where the drift $h(\theta)$ is linear in θ.

Using the SA framework developed earlier, one can show:

- $\theta_n \to \theta^\star$ almost surely,
- $\Phi\theta^\star$ is the unique fixed point of the *projected Bellman operator*.

Conclusion: Linear TD learning converges to the best approximation of V^μ within the feature-induced subspace $\mathcal{V}_\Phi$.

Outline

Next Steps

We have completed our discussion on **policy evaluation**.

In the next lecture, we will shift our focus to **policy optimization**:

- how to improve a policy using value-function estimates,
- the connection between evaluation and control,
- iterative schemes for learning optimal policies.

SERIES WEBPAGE

ml theory series

The ML Theory series, co-organised with Databased, consists of a set of mini-courses covering topics in Theoretical Machine Learning. Each mini-course is complemented by invited talks from faculty members and researchers at industrial research labs, highlighting state-of-the-art developments in the respective areas.

Calendar: Subscribe on Google Calendar

Reinforcement Learning

Date	Lecture Topic	Instructor	Notes
6 Jan 2026	Markov Decision Processes, Value Iteration	Kintan Saha	Slides | Scribe | Recording
8 Jan 2026	Policy Iteration, Temporal Difference Learning	Kintan Saha	Slides | Scribe | Recording
10 Jan 2026	Stochastic Approximation, Convergence of TD	Kintan Saha	Slides | Scribe | Recording
12 Jan 2026	Q Learning, Policy Gradient Methods	Kintan Saha, Ishaq Hamza	Slides | Scribe | Recording
14 Jan 2026	Actor-Critic Methods	Ishaq Hamza	Slides | Scribe | Recording
20 Jan 2026	Multi-Agent Systems - I	Siddharth Reddy	Slides | Scribe | Recording
22 Jan 2026	Multi-Agent Systems - II	Siddharth Reddy	Slides | Scribe | Recording

Thank You

Thank you for your attention!

Questions, comments, or clarifications?