Approximate value iteration and value function approximations

Daniel Russo

March 30, 2020

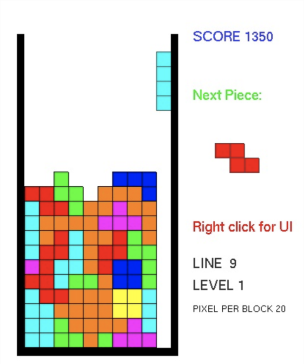
Columbia University

Readings for today

- Temporal difference learning is treated thoroughly in the textbook Sutton and Barto [2018]. Our textbook Bertsekas [1995] provides a more linear algebraic perspective in Section 6.3 of Volume II, 4th edition.
- Mnih et al. [2015] applies Q-learning with deep neural networks to Atari games. This launched a resurgence of interest in RL.
- The convergence analysis presented here was mostly discovered by Tsitsiklis and Van Roy [1997] and Munos and Szepesvári [2008].

Table of Contents

Toy motivation: Value prediction in tetris



- State $s \in \{0,1\}^{10\times 20}$ is a board configuration.
- Observe a given algorithm play repeated games.
- Goal is to total reward accrued over the "near" future.
- One approach is to fit a linear approximation: $J^{\mu}(s) \approx J_{\theta}(s) := \phi(s)^{\top}\theta$.
- $\phi(s)$ encodes features. E.g. column heights, inter-column height differences, max height etc.

A single possession value functions in sports analytics

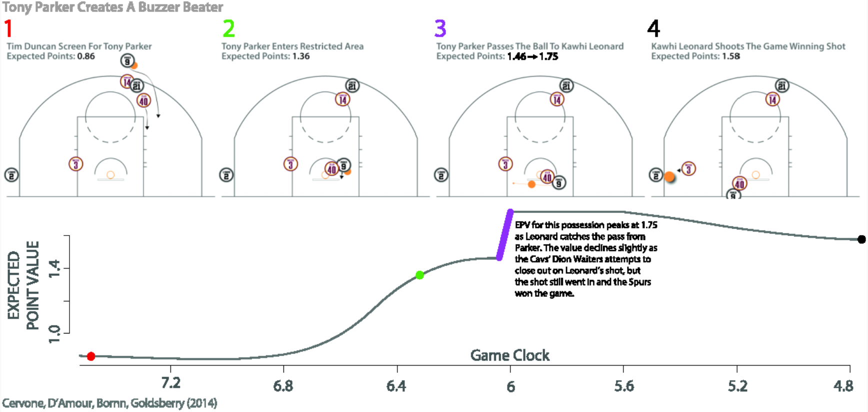
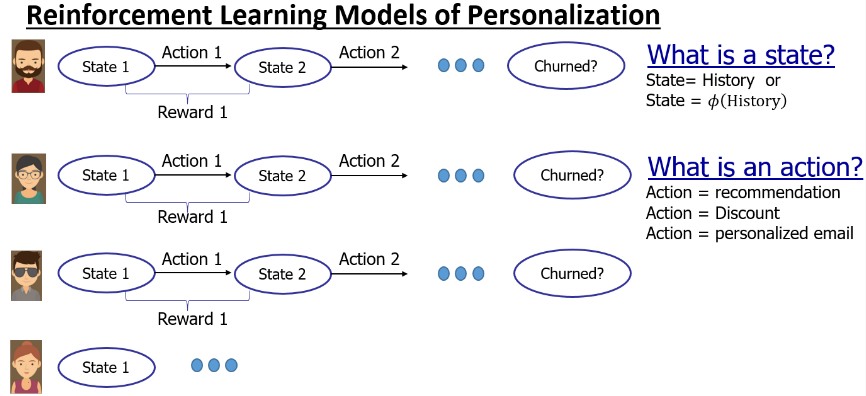


Figure 2. EPV throughout the Spurs' final possession, with annotations of major events.



$J^{\mu}(s)$ captures net present value derived from a customer in "state" s under the status-quo policy π.

Outline for this section

1. How should we estimate J^{μ} using the features $\phi(\cdot)$?
   - *Monte carlo value function approximation:*
     Each episode gives a single sample of $J^{\mu}(s_0)$
   - *Temporal difference (TD) methods:*
     Each state transition gives an observation of the error in Bellman's equation. Aim to minimize temporal inconsistency.
2. Convergence of TD type methods with linear function approximation and on policy sampling.

When we study policy iteration and approximate policy iteration, we will see precisely how cost-to-go approximations are used to produce improved policies.

Setup

- We are observe a Markovian sequence $s_0, s_1, \cdots, s_n, \cdots$ on a finite state space $\mathcal{S} = \{1, \cdots, |\mathcal{S}|\}$ that is generated by applying policy μ.
- Assume the Markov chain under μ is irreducible and aperiodic. It therefore has unique stationary distribution π, i.e. $\pi = \pi P_\mu$, and $\mathbb{P}(s_n = s | s_0) \to \pi(s)$.
    - For practical results, we also need that this Markov chain mixes "rapidly."
- For convenience, assume the chain is stationary. That is, $s_0 \sim \pi$ so $s_t \sim \pi$ for each t. This is akin to assuming we throw away the first τ_{mix} state observations.

Setup (continued)

- Focus on linear value function approximation $J_\theta(s) = \phi(s)^\top \theta$.
- Can write $J_\theta = \Phi\theta$ where $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$ with rows $\Phi_s = \phi(s)^\top$.
- Assume the feature covariance matrix is non-degenerate: For $S \sim \pi$,

$$\mathbb{E}\left[\left(\phi(S) - \mathbb{E}[\phi(S)]\right)\left(\phi(S) - \mathbb{E}[\phi(S)]\right)^\top\right] \succ 0$$

.

Table of Contents

Monte Carlo value function estimation

For each state s_n, we have a noisy observation of the N step discounted cost:

$$G_{n:n+N} = \sum_{t=n}^{n+N} \alpha^{t-n} g_\mu(s_t)$$

Except for the truncation at N periods, this gives an unbiased estimate of the cost-to-go. In particular:

$$\mathbb{E}\left[G_{n:n+N} | s_n\right] = J^\mu(s_n) - \underbrace{\mathbb{E}\left[\sum_{t=N+1}^{\infty} \alpha^t g_\mu(s_t) \mid s_n\right]}_{O(\frac{\alpha^N}{1-\alpha})}. \qquad (1)$$

Given $n + N$ state observations, we can estimate $J^\mu(s)$ as $J_{\theta_n^{\mathrm{MC}}}(s) = \phi(s)^\top \theta_n^{\mathrm{MC}}$ where

$$\theta_n^{\mathrm{MC}} = \operatorname*{argmin}_\theta \sum_{i=0}^{n} \left(\phi(s_i)^\top \theta - G_i\right)^2.$$

Monte Carlo estimation asymptotics

Taking $n \to \infty$ and then $N \to \infty$, we get

$$\theta_n^{\mathrm{MC}} \xrightarrow{a.s.} \operatorname*{argmin}_{\theta} \sum_{s \in \mathcal{S}} \pi(s) \left(\phi(s)^\top \theta - J^\mu(s) \right)^2 .$$

In terms of cost-to-go-functions, taking $n \to \infty$ and then $N \to \infty$

$$J_{\theta_n^{\mathrm{MC}}} \xrightarrow{a.s.} \Pi_\pi J^\mu$$

where $\Pi_\pi J = \operatorname{argmin}_{\hat{J} \in \mathrm{Col}(\Phi)} \|\hat{J} - J\|_{2,\pi}$ is the projection on the space spanned by the feature vectors in the π weighted 2-norm.

For the curious, for fixed N, as $n \to \infty$,

$$J_{\theta_n^{\mathrm{MC}}} \xrightarrow{a.s.} \Pi_\pi T_\mu^N \vec{0}$$

Temporal difference methods: Least-Squares Temporal Difference Learning (LSTD)

Another idea is to approximate a Bellman iteration within the span of our features. We generate a sequence of estimates $\left(\hat{\theta}_1, \cdots, \hat{\theta}_K\right)$, by solving for an approximate Bellman update:

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(J_\theta(s_i) - \underbrace{[g_\mu(s_i) + \gamma J_{\theta_k}(s_{i+1})]}_{T_\mu J_{\theta_k}(s_i) + \text{noise}} \right)^2$$

In RL lingo, this method "bootstraps," because it uses its current estimate $J_{\theta_k}(s_{i+1})$ as a learning target.

Asymptotics of temporal difference methods

Taking the number of observations $n \to \infty$, and *then*, LSTD becomes the projected Bellman iteration

$$J_{\theta_{k+1}} = \Pi_\pi T_\mu J_{\theta_k} \ \ k = 1 \cdots K - 1.$$

Assuming for now (we will return to study this) that converges as $K \to \infty$, it should converge to the so-called TD fixed point

$$J_{\mathrm{TD}} = \Pi_\pi T_\mu J_{\mathrm{TD}}.$$

Since Π_π and T_μ are linear, this is a linear point fixed equation.

Questions:

1. What are the advantages of TD vs MC?
2. Does this converge as $K \to \infty$?
3. Can we give a guarantee on the quality of the TD fixed point?

Sidenote: Online temporal difference learning

Instead of the full updates

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(J_\theta(s_i) - \underbrace{[g_\mu(s_i) + \gamma J_{\theta_k}(s_{i+1})]}_{T_\mu J_{\theta_k}(s_i) + \text{noise}} \right)^2,$$

we can run a fully online verison of TD

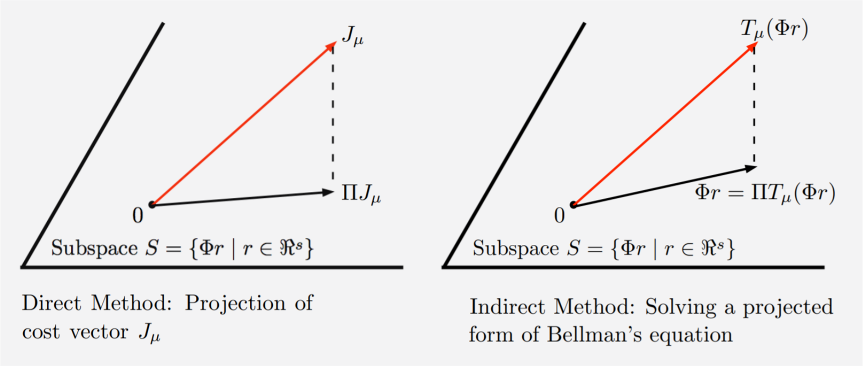
$$\theta_{i+1} = \theta_i - \gamma \nabla_\theta \frac{1}{2} \left(J_\theta(s_i) - [g_\mu(s_i) + \gamma J_{\theta_i}(s_{i+1})] \right)^2 \Big|_{\theta=\theta_i}$$

One of the central, most distinctive, ideas in reinforcement learning. See Sutton and Barto [2018] for a thorough introduction.

Does this circular process converge? The classic convergence theory is due to Tsitsiklis and Van Roy [1997]. A fairly clean finite time analysis is given in Bhandari et al. [2018].

Visualizing the difference between TD and MC

Picture taken from Dimitris Bertsekas. (Replace $r \equiv \theta$)



The method on the left is what we're calling Monte Carlo. The method on the right is what we're calling TD-type methods.

Temporal difference methods vs Monte Carlo methods

Which is better? In which circumstances?

It is unclear to me if anyone fully understands, but ...

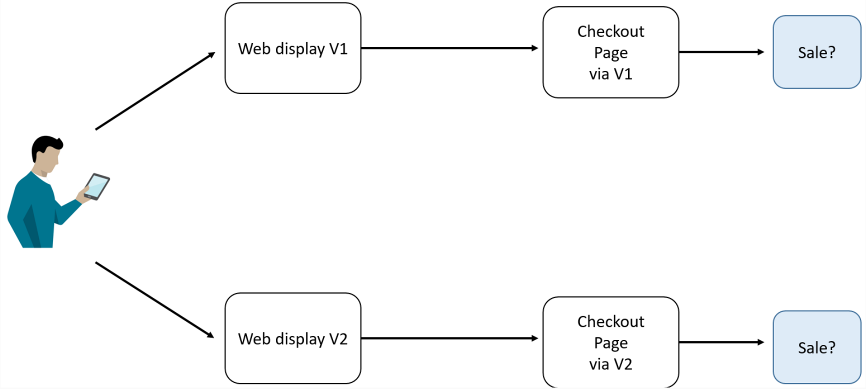
- MC methods directly minimize the "true" loss function, but can have high variance when the horizon is long.
  - Notice that the MC estimator compresses the entire state trajectory into a single number $G_{n:n+N}$, losing vital information.
- TD methods can have much lower variance, but introduce bias.

An example

We want to estimate the success rate of two ads. We observe

1. whether they click
2. whether they subsequently complete purchase

The conversion rate from click to sale is low and hard to estimate.



- This problem has four states

$$\mathcal{S} = \{\text{Web V1, Web V1, Checkout V1, Checkout V2}\}$$

- We select features the state aggregation features:

$$\phi(\text{Web V1}) = e_1 \qquad \phi(\text{Checkout V1}) = e_3$$
$$\phi(\text{Web V2}) = e_2 \qquad \phi(\text{Checkout V1}) = e_3$$

- The MC estimator is simple average:
  - We estimate the sale rate of each display, θ_1^{MC} and θ_2^{MC}, to be the proportion of customers who *purchased* the item upon seeing that ad.
- Under TD, we pool data:
  1. First estimate θ_3^{TD} to be the proportion of sales among customers who reach the checkout page, regardless of how they reach the checkout.
  2. Then estimate θ_1^{TD} to be proportion of customers who click on V1 times the the estimated sale probability after clicking, θ_3^{TD}.

Interpolating between TD and MC using m step returns

A tuning parameter m to interpolate between TD and MC...

We generate a sequence of estimates $\left(\hat{\theta}_1, \cdots, \hat{\theta}_K\right)$, by solving for an m step Bellman update:

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(J_\theta(s_i) - \underbrace{\left[G_{i:i+m} + \gamma J_{\theta_k}(s_{i+m+1})\right]}_{T_\mu^m J_{\theta_k}(s_i) + \text{noise}} \right)^2$$

Taking the number of observations $n \to \infty$, and *then*, this becomes the projected Bellman iteration

$$J_{\theta_{k+1}} = \Pi_\pi T_\mu^m J_{\theta_k} \;\; k = 1 \cdots K-1.$$

Assuming for now (we will return to study this) that this converges as $K \to \infty$, it should converge to the fixed point

$$J_{\mathrm{TD(m)}} = \Pi_\pi T_\mu^m J_{\mathrm{TD(m)}}.$$

Table of Contents

Convergence to a TD fixed point

Fact Euclidean projections are non-expansions:
$\|\Pi_\pi J\|_{2,\pi} \leq \|J\|_{2,\pi}$.

Lemma *T_μ is a contraction in $\|\cdot\|_{2,\pi}$ with modulus α.*

Proof: Let $S = s_t, S' = s_{t+1}$ for any t.
By stationarity $\mathbb{P}(S = s) = \mathbb{P}(S' = s) = \pi(s)$ for each s.
But, these are dependent, with $\mathbb{P}(S' = s' | S = s) = P_\mu(s, s')$.
Note that $T_\mu J(S) = g_\mu(S) + \gamma \mathbb{E}[J(S') \mid S]$.

$$
\begin{aligned}
\|T_\mu J - T_\mu \bar{J}\|_{2,\pi} &= \sqrt{\mathbb{E}\left[\left(T_\mu J(S) - T_\mu \bar{J}(S)\right)^2\right]} \\
&= \sqrt{\mathbb{E}\left[\left(\gamma \mathbb{E}\left[J(S') - \bar{J}(S') \mid S\right]\right)^2\right]} \\
&\leq \gamma \sqrt{\mathbb{E}\left[\left(J(S') - \bar{J}(S')\right)^2\right]} = \gamma \|J - \bar{J}\|_{2,\pi}.
\end{aligned}
$$

Convergence to a TD fixed point

The previous slides shows $\Pi_\pi T_\mu$ is a contraction, i.e.

$$\|\Pi_\pi T_\mu J - \Pi_\pi T_\mu \bar{J}\|_{2,\pi} \leq \|T_\mu J - T_\mu \bar{J}\|_{2,\pi} \leq \alpha \|J - \bar{J}\|_{2,\pi}.$$

This critically relies on the fact that the state-relevance weighting is the stationary distribution under the policy μ.

What is the TD fixed point?

What does it mean that $J_{\mathrm{TD}} = \Pi_\pi T_\mu J_{\mathrm{TD}}$?

- This means that errors in Bellman's equation are orthogonal to the features, i.e. in the inner product $\langle J, J' \rangle_\pi = \sum_s \pi(s) J(s) J'(s)$ we have

$$\langle \Phi_{:,i}, J_{\mathrm{TD}} - T_\mu J_{\mathrm{TD}} \rangle_\pi = 0 \quad i = 1, \cdots, d$$

- $\hat{\theta}_{\mathrm{TD}}$ minimizes the mean-squared Projected Bellman error

$$\mathrm{MSPBE}(\theta) = \|\Pi_\pi (J_\theta - T_\mu J_\theta)\|_{2,\pi}$$

See Sutton et al. [2009] for more.

Quality of a TD fixed point

The main result is that TD cannot amplify error *too much* relative to the Monte-carlo estimator. In practice, we hope the factor of $\frac{1}{1-\alpha}$ is quite conservative.

Lemma: $\|J_{\mathrm{TD}} - J_\mu\|_\pi \leq \sqrt{\frac{1}{1-\alpha}} \min_\theta \|J_\theta - J_\mu\|_{2,\pi}$
Proof: Denote $T = T_\mu$, $\Pi = \Pi_\pi$ and $\|\cdot\| = \|\cdot\|_{2,\pi}$. Then, by the Pythagorean theorem,

$$\begin{aligned}
\|J_{\mathrm{TD}} - J_\mu\|^2 &= \|J_{\mathrm{TD}} - \Pi J_\mu\|^2 + \|J_\mu - \Pi J_\mu\|^2 \\
&= \|\Pi T J_{\mathrm{TD}} - \Pi T J_\mu\|^2 + \|J_\mu - \Pi J_\mu\|^2 \\
&\leq \alpha^2 \|J_{\mathrm{TD}} - J_\mu\|^2 + \|J_\mu - \Pi J_\mu\|^2
\end{aligned}$$

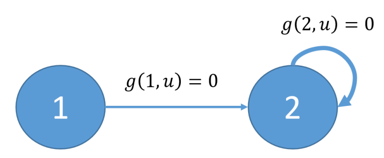
Rearrange terms and use $1/\sqrt{1-\alpha^2} \leq 1/\sqrt{(1-\alpha)}$.

Table of Contents

Divergence with off policy sampling

Two states, 1 action, $J_\mu = (0, 0)$.

Simple function class $J_\theta = (1, 2)\theta$, so $J_0 = J_\mu$.



Divergence

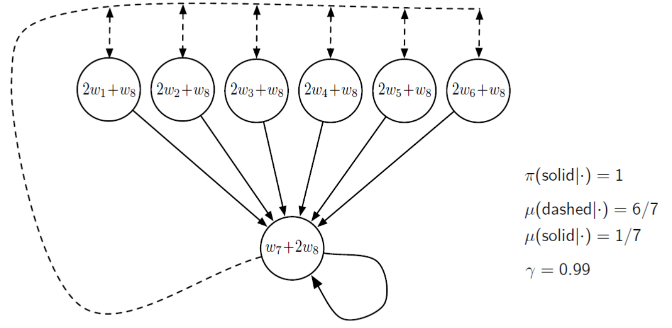
$$\begin{aligned}
\theta_{k+1} &= \operatorname*{argmin}_\theta \|J_\theta - T_\mu J_{\theta_k}\|^2_{\nu,2} \\
&= \operatorname*{argmin}_\theta \nu(1)\left(J_\theta(1) - \gamma J_{\theta_k}(2)\right)^2 + \nu(2)\left(J_\theta(2) - \gamma J_{\theta_k}(2)\right)^2 \\
&= \operatorname*{argmin}_\theta \nu(1)\left(\theta - 2\gamma\theta_k\right)^2 + \nu(2)\left(2\theta - 2\gamma\theta_k\right)^2 \\
&= 2\gamma\left[\frac{\nu(1) + \nu(2)}{\nu(1) + 2\nu(2)}\right]
\end{aligned}$$

If $\gamma > 1/2$ and $\nu(1)/\nu(2)$ is sufficiently close to 1, then $\theta_k \to \infty$.

Divergence with off policy sampling

A richer example, due to Baird, also involves actions.

This figure is from Sutton and Barto [2018], who provides a discussion.

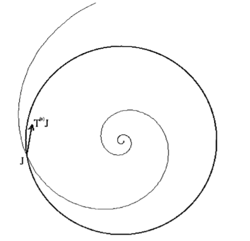


Divergence with nonlinear function approximation

This figure is from Tsitsiklis and Van Roy [1997], They construct a family of functions

$$\theta \in \mapsto J_\theta \in \{J \in \mathbb{R}^3 | J(1) + J(2) + J(3) = 0\}$$

that forms the spiral below. TD dynamics follow the spiral.



Cai et al. [2019] overcome this for Neural networks in the "neural tangent kernel" regime.
Brandfonbrener and Bruna [2019] overcomes this for homogenous function approximators, including ReLU networks.

Table of Contents

From Last Class: fitted value iteration (FVI)

Regression based approximation to Bellman updates.

- Define the weighted norm $\|J\|_{2,\nu} = \sqrt{\mathbb{E}_{s\sim\nu}[J(s)^2]}$.
- Fitted value iteration is the scheme: for $k = 1, 2, \cdots$

$$\theta_{k+1} \in \operatorname*{argmin}_{\theta\in\Theta} \|J_\theta - TJ_{\theta_k}\|_{2,\nu}$$

- Equivalently, this can be viewed as a projected value iteration:

$$J_{k+1} = \Pi_{\mathcal{F},\nu} TJ_k$$

where $\mathcal{F} = \{J_\theta \mid \theta \in \Theta\}$ is the space of value functions approximations and $\Pi_{\mathcal{F},\nu} J = \operatorname{argmin}_{f\in\mathcal{F}} \|f - J\|_{2,\nu}$ projects onto $\mathcal{F}$ in a weighted norm.

From last class: fitted Q-iteration

It is simpler to approximate Bellman updates to Q-functions.

- Define the state-action value function:

$$Q^*(s,u) = g(s,u) + \alpha \sum_{s' \in S} P_{ss'}(u) J^*(s')$$

- Obeys the Bellman $Q^* = FQ^*$ where

$$FQ(s,u) := g(s,u) + \alpha \sum_{s' \in S} P_{ss'}(u) \min_{u'} Q(s',u')$$

- Fitted Q iteration is the scheme: for $k = 1, 2, \cdots$

$$\theta_{k+1} \in \underset{\theta \in \Theta}{\operatorname{argmin}} \| Q_\theta - FQ_{\theta_k} \|_{2,\nu} \tag{2}$$

where ν is a distribution over state, control pairs.

Q-learning

Given a collection of data $\mathcal{D} = \{(s, u, c, s')\}$ where (c, s') denote the cost upon selecting control u in state s.

Fitted Q iteration is

$$\theta_{k+1} = \operatorname*{argmin}_{\theta} \underbrace{\mathbb{E}_{(s,u,c,s')\sim\mathcal{D}} \left[\left(Q_\theta(s, u) - \left(c + \gamma \max_{u'} Q_{\theta_k}(s, u') \right) \right)^2 \right]}_{\mathcal{L}(\theta|\theta_k,\mathcal{D})}$$

Q-learning (with "experience replay") is the iteration

1. Sample small minibatch $\hat{\mathcal{D}}_k \subset \mathcal{D}$.
2. Update

$$\theta_{k+1} = \theta_k - \gamma \nabla_\theta \mathcal{L}(\theta|\theta_k, \hat{\mathcal{D}}_k) \Big|_{\theta=\theta_k}$$

Human-level control through deep reinforcement learning

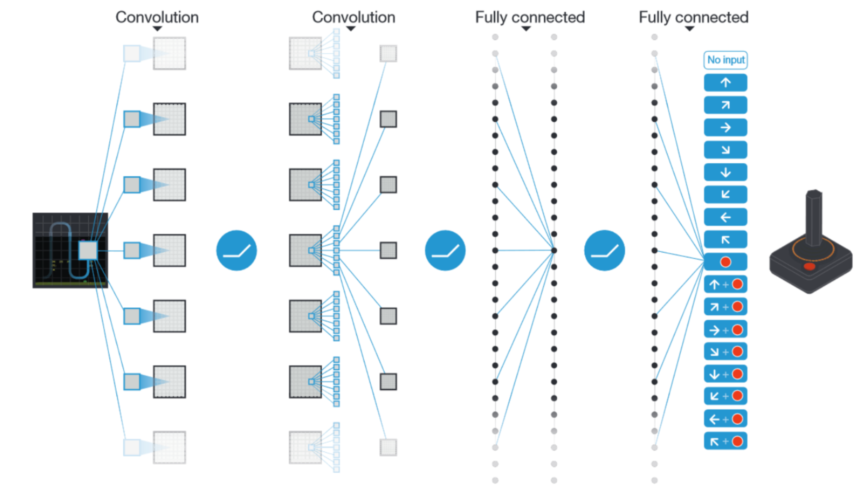


Table of Contents

Divergence of fitted value iteration

We'll focus on fitted value iteration

$$J_{k+1} = \Pi_{\mathcal{F},\nu} T J_k \quad k = 1, 2, \cdots$$

but the same steps apply for Q functions.

Bad news: Bertsekas and Tsitsiklis [1996] gives an example where $J_k \to \infty$ with one dimensional linear function approximation.

Reason for non-convergence

T is a contraction in $\|\cdot\|_\infty$

But $\Pi_{\mathcal{F},\nu}$ could be an expansion in $\|\cdot\|_\infty$

- Consider $\mathcal{S} = \{1, 2\}$, $\nu = (.75, .25)$, $\mathcal{F} = \{(1, 2)\theta : \theta \in \mathbb{R}\}$
- Take $J = (2, 1)$. Then, solving

  $$\operatorname*{argmin}_\theta .75(\theta - J(1))^2 + .25(2\theta - J(2)^2 = \frac{3.5}{2}$$

  gives $\|\Pi_{\nu,\mathcal{F}} J\|_\infty = 3.5$.

Then why does this often work?

- The approximate DP literature is full of examples of divergence with simple linear function approximators.
- Perhaps we should expect much better behavior with universal function approximators
    - non-parametric function classes or over-parameterized neural-networks (which are effectively non-parametric)

Inherent Bellman Error

Define the inherrent Bellman error of the function class:

$$\epsilon = \sup_{J \in \mathcal{F}} \inf_{\hat{J} \in \mathcal{F}} \|\hat{J} - TJ\|_\infty = \sup_{J \in T\mathcal{F}} \inf_{\hat{J} \in \mathcal{F}} \|\hat{J} - J\|_\infty$$

If $\epsilon = 0$, then $\mathcal{F}$ is *closed under Bellman updates*.

Define the approximate Bellman operator $\hat{T}J = \Pi_{\mathcal{F},\nu} TJ$.

Then

$$\sup_{J \in \mathcal{F}} \|\hat{T}J - TJ\|_\infty = \epsilon$$

Your homework #6 shows

$$\limsup_{k \to \infty} \|J_k - J^*\|_\infty \le \frac{\epsilon}{1 - \alpha}.$$

Inherent Bellman Error

Is the inherent Bellman error small with expressive function classes?

- Yes. If $\mathcal{F} = \mathbb{R}^{|\mathcal{S}|}$ is all cost-to-go functions, it is zero.
- No. As $\mathcal{F}$ gets richer, so does $T\mathcal{F} = \{TJ : J \in \mathcal{F}\}$.
  - Chasing your own tail.

In reality, with non-parametric function classes, we start with a simple J_θ and complexity increases as the number of iterations (and hence data in a fully online procedure) increases. To my understanding, this is not captured at all by current theory.

Overview of results of Munos and Szepesvári [2008]

- The term inherent Bellman error is due to Munos and Szepesvári [2008]. The analysis in max-norm was completed much earlier by Bertsekas and Tsitsiklis [1996].
- Munos and Szepesvári [2008] work in more general euclidean norms and more carefully bound finite sample error.

In our setting, results of Munos and Szepesvári [2008] imply

$$\limsup_{k\to\infty} \|J^{\mu_k}(d_0) - J^*\|_{2,\nu} \leq \frac{\sqrt{C_\nu}}{1-\alpha} \inf_{J\in\mathcal{F}} \inf_{\hat{J}\in\mathcal{F}} \|\hat{J} - TJ\|_{2,\nu}$$

where

$$C_\nu \approx \sup_{\mu} \left\| \frac{d_\infty^\mu}{\nu} \right\|_\infty$$

is small if ν appropriately weights the states visited by *any* policy.

Dimitri P Bertsekas. Dynamic programming. *Deterministic and Stochastic Models*, 1995.

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. *arXiv preprint arXiv:1806.02450*, 2018.

David Brandfonbrener and Joan Bruna. Geometric insights into the convergence of nonlinear td learning. *arXiv preprint arXiv:1905.12185*, 2019.

Qi Cai, Zhuoran Yang, Jason D Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. In *Advances in Neural Information Processing Systems*, pages

11312–11322, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May): 815–857, 2008.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora.

Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000, 2009.

John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 42(5), 1997.