Chapter 11

Off-policy methods with approximation

Recall off-policy learning involves two policies

- - the *target policy*
- Another policy μ that is used to select actions
	- the *behavior policy*

• One policy *π* whose value function we are learning

Off-policy is much harder with Function Approximation

- Even linear FA
- Even for prediction (two fixed policies π and μ)
- Even for Dynamic Programming
- The deadly triad: FA, TD, off-policy
	- Any two are OK, but not all three
	- With all three, we may get instability (elements of θ may increase to $\pm \infty$)

There are really 2 off-policy problems One we know how to solve, one we are not sure One about the future, one about the present

• The easy problem is that of off-policy targets (future)

- We have been correcting for that since Chapters 5 and 6
- Using importance sampling in the target
- The hard problem is that of the distribution of states to update (present); we are no longer updating according to the on-policy distribution

Baird's counterexample illustrates the instability 244 *CHAPTER 11. OFF-POLICY METHODS WITH APPROXIMATION*

What causes the instability?

- It has nothing to do with learning or sampling
	- Even dynamic programming suffers from divergence with FA
- It has nothing to do with exploration, greedification, or control
	- Even prediction alone can diverge
- It has nothing to do with local minima or complex non-linear approximators
	- Even simple linear approximators can produce instability

The deadly triad

• significantly generalizing from large numbers of examples

- - 1. Function approximation
		-
	- 2. Bootstrapping
		-
	- 3. Off-policy learning
		-

• learning value estimates from other value estimates, as in dynamic programming and temporal-difference learning

• learning about a policy from data not due to that policy, as in Q-learning, where we learn about the greedy policy from data with a necessarily more exploratory policy

(Why is dynamic programming off-policy?)

Any 2 Ok

• The risk of divergence arises whenever we combine three things:

TD(0) can diverge: A simple example TD update: $\Delta\theta$ TD fixpoint: $\theta^* = 0$ θ 20 $\delta = r$ $= 0$ $=$ θ $= \alpha \delta \phi$ $= \alpha \theta$ Diverges!

$$
-Q\left(\frac{1}{2}\theta\right)
$$

$$
+\gamma\theta^\top\phi'-\theta^\top\phi\\+\ 2\theta-\theta
$$

Can we do without bootstrapping?

- Bootstrapping is critical to the computational efficiency of DP
- Bootstrapping is critical to the data efficiency of TD methods
- On the other hand, bootstrapping introduces bias, which harms the asymptotic performance of approximate methods
- The degree of bootstrapping can be finely controlled via the λ parameter, from $\lambda=0$ (full bootstrapping) to $\lambda=1$ (no bootstrapping)

4 examples of the effect of bootstrapping suggest that $\lambda=1$ (no bootstrapping) is a very poor choice

Desiderata: We want a TD algorithm that

- Bootstraps (genuine TD)
- Works with linear function approximation (stable, reliably convergent)
- Is simple, like linear TD $O(n)$
- Learns fast, like linear TD
- Can learn off-policy
- Learns from online causal trajectories (no repeat sampling from the same state)

4 easy steps to stochastic gradient descent

- 1. Pick an objective function $J(\theta)$, a parameterized function to be minimized
- 2. Use calculus to analytically compute the gradient $\nabla_\theta J(\theta)$
- 3. Find a "sample gradient" $\nabla_{\theta} J_t(\theta)$ that you can sample on every time step and whose expected value equals the gradient
- 4. Take small steps in θ proportional to the sample gradient:

 $\theta \leftarrow \theta - \alpha \nabla_{\theta} J_t(\theta)$

$$
\Delta \theta = \alpha \delta \phi
$$

$$
\delta = r + \gamma \theta^{\top} \phi' - \theta^{\top} \phi
$$

$$
\mathfrak{t}^{\mathfrak{t}}\mathfrak{c}
$$

Conventional TD is not the gradient of anything

$$
\frac{\partial^2 J}{\partial \theta_j \partial \theta_i} = \frac{\partial (\delta \phi_i)}{\partial \theta_j} = (\gamma \phi'_j - \phi_j) \phi_i
$$

$$
\frac{\partial^2 J}{\partial \theta_i \partial \theta_j} = \frac{\partial (\delta \phi_j)}{\partial \theta_i} = (\gamma \phi'_i - \phi_i) \phi_j
$$

Real 2nd derivatives must be symmetric

$$
\frac{\partial J}{\partial \theta_i} = \delta \phi_i
$$

Then look at the second derivative:

TD(0) algorithm:

Assume there is a J such that

$$
\frac{\partial^2 J}{\partial \theta_j \partial \theta_i} \neq \frac{\partial^2 J}{\partial \theta_i \partial \theta_j}
$$

Etienne Barnard 1993

A-split example (Dayan 1992)

$V(A)=0.5$ $V(B)=1$ Clearly, the true values are

 $\overline{}$ $\begin{array}{c} \bigcup_{i=1}^n A_i \cup B_i \cup B_i \end{array}$ But if you minimize the naive objective fn,

1 get the sc $J(\theta) = \mathbb{E}[\delta^2]$, then you get the solution

Even in the tabular case (no FA)

$$
V(A)=1/3
$$

 $V(B)=2/3$

Now we must finish this section by discussing the relative merits of the second and Now we must finish this section by discussing the relative merits of the second and $J_\text{RE}(\theta)^2=J_\text{VE}(\theta)^2+\mathbb{E}\left[\left(v_\pi(\theta)\right)^2\right]$ one action (or, equivalently, no action (or, equivalently, no action (or, equivalently, no action (or, equivalently, α α of α states in the state α $\sqrt{2\pi}$ on the equation of the edge indicate indicate indicate indicate indicate in \mathcal{L} defined distinctly; each has a separate that are represented distinctly; each has a separate on the separate that are represented distinctly in the separate that are represented distinctly in the separate of \mathcal $\frac{1}{\tau}$ can $\frac{1}{\tau}$ can $\frac{1}{\tau}$ can $\frac{1}{\tau}$ can $\frac{1}{\tau}$ can $\frac{1}{\tau}$ can be right has the ri $\sigma_{\mathrm{KE}}(v) = \sigma_{\mathrm{VE}}(v)$ in α α α α α α but the semal Deturn α \bigvee σ that they can take on σ take on σ take on σ the σ $\mathcal{O}_{\text{KL}}(\nu) = \mathcal{O}_{\text{VL}}(\nu) + \mathcal{Q}_{\text{KL}}(\nu)$ channels. Where two hour different V probability. The numbers on the edges indicate the reward emitted if that edge is traversed. $\frac{1}{2}$ of $\frac{1}{2}$ but the same Return Errors $\frac{1}{\gamma}$ weight so that they can take on $\frac{1}{\gamma}$ $\sqrt{UUUUUUU}$ and \sqrt{UUUUU} and \sqrt{UUUU} $J_{\rm RE}(\theta)^2=J_{\rm VE}(\theta)^2+{\mathbb E}\left[\left(v_\pi(S_t)-G_t\right)^2\;|\;x\right]$ the value of B 0 is given by the second. Notice that the second. Notice that the observable data is identically data is identically defined at Γ T_{beam} two edges leaves at f_{beam} to f_{beam} $p = \frac{1}{2}$ on the se two have different Value Errors, $\frac{1}{\sqrt{2}}$ we can take on the motion take on the motion $\frac{1}{\sqrt{2}}$ (both errors have the same minima) $I_{\text{DE}}(\theta)^2 = I_{\text{VE}}(\theta)^2 + \mathbb{E} \left[\left(v_1(S_t) - G_t \right)^2 \middle| A_t \sim \pi \right]$ $\sum_{i=1}^n$ is given by the second $\sum_{i=1}^n$ is in identical is in identical is in identical is in identical in ide where \mathbf{r} the returns would be under the target policy: \mathbf{r} ho came $\overline{}$ R **at**₁ Irrn \overline{C} $rros$ \overline{a} $J_\mathrm{RE}(\theta)^2 = J_\mathrm{VE}(\theta)^2 + \mathbb{E}$ \lceil $v_{\pi}(S_t) - G_t$ $\left| \right\rangle ^{2}$ $\big| A_{t:\infty} \sim \pi$ $\overline{}$

> 0, then some number of Bs each followed by a 1, except the last which is followed by a 1, the set wo have different and a single A and a single the errors have different and the second the second MDP, the solution produces and B0 of 1, \sim \mathbf{r} $\text{(A)} \quad \text{(B)} \quad \text{L} \quad \text{(B)} \quad \text{h}}$ is the come Droiente $\bigcup_{n\in\mathbb{N}}$ as well; in both MDPs, the probability of a string of *k* Bs is 2*k*. Now consider the value $\sum_{n=0}^{\infty}$ and $\sum_{n=0}^{\infty}$ increased the induction, and the overall BE is $\sum_{n=0}^{\infty}$ $t(A)$ (B) \leftarrow (B) and $B)$ but the same Projected Be), or p2*/*3 if the three states are equally weighted by *d*. The two MDPs, where the same data, which generates the same design be estimated from the BES. The BES as well; in both MDPs, the probability of a string of *k* Bs is 2*k*. Now consider the value ⁻¹ In the first MD paye different Bellma the second MDP, this solution produces an error in both B and B0 of 1, for an overall BE $\overline{P}(B)$ but the same Projected Bellman Errors \forall (the errors have different minima) These two have different Bellman Errors,

Indistinguishable pairs of MDPs they also are not identifiable and cannot be determined from *Pµ*(⇠) alone. The possible dependency relationships among the data distribution, MDPs, and var-The minimum is achieved at the *projection fixpoint*, at which ious objectives are summarized in Figure 4. The left side of the figure treats the non*s*2*S* The minimum is achieved at the *projection fixpoint*, at which *s*2*S* The minimum is achieved at the *projection fixpoint*, at which estimate it from data. One of the simplest examples is the pair of MDPs shown below: 1 -1 estimate it from data. One of the simplest examples is the pair of MDPs shown below: IV pu **ngirc** AII V 1 -1 data produced by two di↵erent MDPs is identical in every respect, yet the BE is di↵erent. In iuruotii iyului di↵erent MDPs can produce the same data distribution. For non-bootstrapping objectives, the VE can be di↵erent for the two MDPs, and thus is not identifiable, but the optimal weights are the same and can be determined by optimizing the

<u>јгу</u> 1 -1 left MDP is 0 while the *J*VE of the right MDP is 1, for any *d*. Thus, the *J*VE is di↵erent for two MDPs with the same data distribution and the *J*VE cannot be determined from data. There is a saving grace, however, however, however, however, the value of two *JVEs can be diverse* two *JVEs can* be diverse two *JVEs can be diverse* two *JVEs can be diverse two* JVEs can be diverse two *JVEs can be di*

Not all objectives can be estimated from data Not all minima can be found by learning IDIGON AGS jectories, and the objectives of learning. As already described, the MDP and behav $t \sim d$ from Let us consider more carefully the relationship between the MDP, the possible data trajectories, and the objectives of learning. As already described, the MDP and behavior policy together completely determine the probability distribution over data trajec-

ata produced by the manimum of the Bellman Frrom In such a case the BE is literally not a function of the data, and the data, and the data, and the data, and α find the minimum of the Bellman Frror In such a case the BE is literally not a function of the data, and the data, and thus the data, and thus the d
, and the data, and thus the data, and the data, and the data, and thus to way to data, and the data, and the In such a case the BE is literally not a function of the data, and thus there is no way to minimum of the Bellman Error In such a case the BE is literally not a function of the data, and thus the data, and thus the data, and to way to the data, and thus to way to w No loorning algorithm aan find the minimum of the Dellmen Error bootstrapping (left) and bootstrapping (right) objectives. In both cases, two No learning algorithm can find the minimum of the Bellman Error

 $\frac{1}{4}$

The Gradient-TD Family of Algorithms

- True gradient-descent algorithms in the Projected Bellman Error
- GTD(λ) and GQ(λ), for learning V and Q
- Solve two open problems:
	- convergent linear-complexity off-policy TD learning
	- convergent non-linear TD
- Extended to control variate, proximal forms by Mahadevan et al.

First relate the geometry to the iid statistics same feature representation; they look the same and must ⇤*D* = (⇥(⇥⇤*D*⇥) 1.
1<mark>.</mark> – Die Sterne Bartham († 1980)
1. – Die Sterne Bartham († 1980) ⇤*D*(⇥(⇥⇤*D*⇥) ¹⇥⇤*D*) **1e geometry to the lid statistic** fine, the approximation the temporal-difference error, $\frac{1}{2}$ ⇤*^k* = *r^k* + ⇥⌅⇧ be representable as *V* for any ⌅. Consider the projection

$\mathbf{MSPBE}(\theta)$ $=$ $\|V_{\theta} - \Pi TV_{\theta}\|_{D}^{2}$ $= \|\Pi(V_{\theta} - TV_{\theta})\|_{D}^{2}$ $= (\Pi(V_{\theta} - TV_{\theta}))^{T} D(\Pi(V_{\theta} - TV_{\theta}))$ $=$ $(V_{\theta} - TV_{\theta})^{\top} \Pi^{\top} D \Pi (V_{\theta} - TV_{\theta})$ $= \mathbb{E}[\delta \phi]$ ['] E $\left[\phi\phi^{\top}\right]^{-1}\mathbb{E}[\delta\phi]$. $\Delta \mathbf{SPE}(\theta)$ and $\Delta \mathbf{PSE}(\theta)$ the proportion of time steps in state *s*. In this case, the sum in (1) \equiv if $V_{\theta} - 11'I'V_{\theta}||_{D}^{2}$ $\begin{array}{ccc} \hline & & & & & \hline & & & & \h$

Derivation of the TDC algorithm

s $\longrightarrow s'$ ϕ ϕ' This is the trick! $w \in \Re^n$ is a second set of weights $\alpha \nabla_{\theta} \parallel V_{\theta} - \Pi TV_{\theta} \parallel_{D}^{2}$ $\sqrt{2}$ $\mathbb{E} \left [\delta \phi \right] \mathbb{E} \left [\phi \phi^\top \right]^{-1} \mathbb{E} \left [\delta \phi \right]$ \setminus $= \ -\alpha \left(\nabla_{\theta} \mathbb{E}\left[\delta \phi\right]\right) \mathbb{E}\left[\phi \phi^{\top}\right]^{-1} \mathbb{E}\left[\delta \phi\right]$ $\overline{ }$ $r + \gamma \phi'$ ' $\theta - \phi$ ' θ \setminus] $\left[\begin{smallmatrix}\phi\phi^\top\end{smallmatrix}\right]^{-1}\mathbb{E}\left[\delta\phi\right]$ $\phi\left(\gamma\phi'-\phi\right)$ T]^{\perp} $\mathbb{E} \left[\phi \phi^\top \right]^{-1} \mathbb{E} \left[\delta \phi \right]$ $= \; -\alpha \left(\gamma \mathbb{E} \left[\phi' \phi^\top \right] - \mathbb{E} \left[\phi \phi^\top \right] \right) \mathbb{E} \left[\phi \phi^\top \right]^{-1} \mathbb{E} \left[\delta \phi \right]$ $\phi^{\prime}\phi^{\top}][\mathbb{E}\left[\phi\phi^{\top}\right]^{-1}\mathbb{E}\left[\delta\phi\right]$ $\phi' \phi^\top \rceil \big| w$

r

• on each transition

• update two parameters TD(0) where, as usual $\theta \leftarrow \theta + \alpha \delta \phi$ $w \leftarrow w + \beta(\delta - \phi)$

with gradient correction

estimate of the TD error (δ) for the current state ϕ

 $\delta = r + \gamma \theta^{\top} \phi' - \theta^{\top} \phi$

⇥

TD with gradient correction (TDC) algorithm aka GTD(0)

 ϕ^+w

Convergence theorems

• All algorithms converge w.p.1 to the TD fix-point:

 α

- - $\mathbb{E}[\delta \phi] \longrightarrow 0$
- GTD, GTD-2 converges at one time scale

• TD-C converges in a two-time-scale sense

$$
\alpha,\beta\longrightarrow 0
$$

$$
\frac{\alpha}{\beta}\longrightarrow 0
$$

$$
\alpha = \beta \longrightarrow 0
$$

Off-policy result: Baird's counter-example

Gradient algorithms converge. TD diverges.

Computer Go experiment

- Learn a linear value function (probability of winning) for 9x9 Go from self play
- One million features, each corresponding to a template on a part of the Go board
- An established experimental testbed

\parallel $\mathbb{E}[\Delta \theta_{TD}] \parallel$

ALGORITHM

Off-policy RL with FA and TD remains challenging; there are multiple ideas, plus combinations In conclusion

-
- Emphatic TD

- Better state rep'ns (less FA)
- Recognizers (less off-policy)
- LSTD (O(n²) methods)

• Gradient TD, proximal gradient TD, and hybrids

More work needed on these novel algs!

Emphatic temporaldifference learning

Reinforcement Learning and Artificial Intelligence Laboratory Department of Computing Science University of Alberta **Canada**

Rupam Mahmood, Huizhen (Janey) Yu, Martha White, Rich Sutton

State weightings are important, powerful, even magical, when using "genuine function approximation" (i.e., when the optimal solution can't be approached)

• They are the difference between convergence and divergence

• They are needed to make the problem well-defined

- in on-policy and off-policy TD learning
-
- more than others in learning

• We can change the weighting by *emphasizing* some steps

Often some time steps are more important

- - Because of *discounting*
	- Because the control objective is to maximize the value of the *starting state*
- In general, function approximation resources are limited
	- Not all states can be accurately valued
	- The accuracy of different state must be traded off!
	- You may want to control the tradeoff

• Early time steps of an *episode* may be more important

Bootstrapping interacts with state importance

• In the Monte Carlo case $(\lambda=1)$ the values of different states (or time steps) are estimated independently,

• But with bootstrapping $(\lambda < 1)$ each state's value is estimated based on the estimated values of later states; if the state is important, then it becomes important to accurately value the later states even if

- and their importances can be assigned independently
- they are not important on their own

Two kinds of importance

- Intrinsic and derived, primary and secondary
	- The one you specify, and the one that follows from it because of bootstrapping
- Our terms: *Interest* and *Emphasis*
	- Your intrinsic *interest* in valuing accurately on a time step
	- The total resultant *emphasis* that you place on each time step

- Data $\cdot\cdot\cdot$ $\boldsymbol{\phi}(S_t)$ A_t R_t .
- State distribution

• Objective to minimize

feature function $\boldsymbol{\phi}: \mathcal{S} \rightarrow \Re^n$

$$
{+1}\stackrel{\centerdot}{\phi}(S{t+1})\;A_{t+1}\;R_{t+2}\;\cdots
$$

interest function target policy $i: \mathcal{S} \rightarrow \Re^+$

importance sampling ratio

$$
d_{\mu}(s) = \lim_{t \to \infty} \Pr
$$

 $\text{MSE}(\boldsymbol{\dot{\theta}}) = \sum d_{\mu}(s)i(s)$ $s \in \mathcal{S}$ parameter vector

• Emphatic TD(0)

$$
(s)i(s)\left(v_{\pi}(s) - \theta^{\top}\phi(s)\right)^{2}
$$

$$
\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha M_t \rho_t \left(R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \right) \boldsymbol{\phi}_t
$$

emphasis $M_t > 0$

$$
\rho_t = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \quad \mathbb{E}[\rho_t] = 1
$$

Problem

Solution

$$
\boldsymbol{\phi}_t = \boldsymbol{\phi}(S_t)
$$

$$
\mathbf{A}_t = \sum_{k=0}^t M_k \rho_k \boldsymbol{\phi}_k \big(\boldsymbol{\phi}_k - \gamma \boldsymbol{\phi}_{k+1}\big)^{\top} \quad \mathbf{b}_t = \sum_{k=1}^t M_k \rho_k R_k \boldsymbol{\phi}_k
$$

$$
\boldsymbol{\theta}_{t+1} = \mathbf{A}_t^{-1} \mathbf{b}_t
$$

\cdots $\phi(S_t)$ A_t R_t .

• State distribution

• Emphatic LSTD(0)

$$
[S_t = s \mid A_{0:t-1} \sim \mu]
$$

$$
\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha M_t \rho_t \left(R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \right) \boldsymbol{\phi}_t
$$
\n
$$
\text{emphasis} \qquad \text{importance sampling ratio}
$$
\n
$$
M_t > 0 \qquad \text{if} \qquad \boldsymbol{\phi}_t
$$

$$
\rho_t = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \quad \mathbb{E}[\rho_t] = 1 \qquad \qquad \boldsymbol{\phi}_t = \boldsymbol{\phi}(S_t)
$$

Problem

- $d_{\mu}(s) = \lim_{t \to \infty}$ $Pr[$
- Objective to minimize parameter vector
	- $\text{MSE}(\boldsymbol{\dot{\theta}}) = \sum d_{\mu}(s)i(s)$ $s \in \mathcal{S}$
- Emphatic TD(0)

Solution

$$
+1 \phi(S_{t+1}) A_{t+1} R_{t+2} \cdots
$$

feature functions and the control of the control o

Emphasis algorithm (Sutton, Mahmood & White 2015)

- Derived from analysis of general bootstrapping relationships (Sutton, Mahmood, Precup & van Hasselt 2014)
- Emphasis is a scalar signal $M_t \geq 0$

$$
M_t = \lambda_t \, i(S_t) + (1 -
$$

$$
F_t = \rho_{t-1} \gamma_t F_{t-1} + i(S_t)
$$

 λ_t *)* F_t

• Defined from a new scalar *followon trace* $F_t \geq 0$

Off-policy implications

- The emphasis weighting is *stable under off-policy TD(*λ*)* (like the on-policy weighting) (Sutton, Mahmood & White 2015)
	- It is the *followon* weighting, from the interest weighted behavior distribution $(d_\mu(s)i(s))$, under the target policy
- Learning is *convergent* (though not necessarily of finite variance) under the emphasis weighting for arbitrary target and behavior policies (with coverage) (Yu 2015)
- There are error bounds analogous to those for on-policy $TD(\lambda)$ (Munos)
- Emphatic TD is the simplest convergent off-policy TD algorithm (one parameter, one learning rate)