SBEED (out of the Deadly Triad): Convergent Reinforcement Learning with Nonlinear Function Approximation

Bo Dai[1→2], Albert Shaw[1], Lihong Li[2], Lin Xiao[3], Niao He[4], Zhen Liu[1], Jianshu Chen[5], Le Song[1]

[1]Gatech, [2]Google Brain, [3]Microsoft Research, [4]UIUC, [5]Tencent AI

Main Results

Background

- This talk: RL $\approx$ solving Bellman equation based on data
- Approx. dynamic programming (DQN, ...) inherently unstable
- Remained challenging for decades — "deadly triad" (Sutton 15)

Contributions

1. Bellman equation reformulated as a saddle-point problem
2. First provably convergent ADP algorithm (SBEED) with general nonlinear function approximation
3. Empirical validation in simulated robotics tasks (Mujoco)

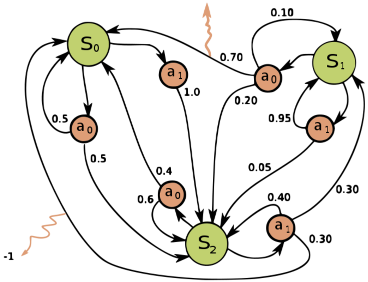
Outline

- **Background**
- Saddle-point Reformulation of Bellman Equation
- SBEED Learning
- Conclusions

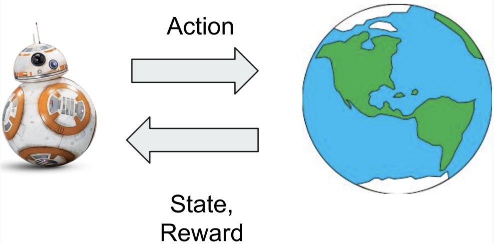
Markov Decision Process (MDP)

$M = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$

- Set of states $\mathcal{S}$
- Set of actions $\mathcal{A}$
- Transition probabilities $P(s'|s, a)$
- Immediate reward $R(s, a)$
- Discount factor $\gamma \in (0, 1)$



(from Wikipedia)



Bellman Equation and Dynamic Programming

Optimal value function V^* satisfies Bellman equation

$$\forall s \in \mathcal{S},\;\; V^*(s) = \underbrace{\max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^*(s')] \right)}_{\mathcal{T}V^*(s)}$$

Well-known facts of Bellman operator $\mathcal{T}$:

- $\mathcal{T}$ is γ-contraction: $\|\mathcal{T}V_1 - \mathcal{T}V_2\|_\infty \leq \gamma \|V_1 - V_2\|_\infty$
- Hence, $V, \mathcal{T}V, \mathcal{T}^2 V, \mathcal{T}^3 V, \cdots \;\rightarrow\; V^*$ ("fixed point")
- Mathematical foundation of value iteration, TD(λ), Q-learning, etc. in the exact ($\approx$ finite-MDP) case

When Bellman Meets Gauss: Approximate DP

In practice, V^* is often approximated

- Eg: least-squares fit on linear models or neural networks, ...
- Composing $\mathcal{T}$ and approximation loses contraction
- Many known divergent examples
  Baird (93), Boyan & Moore (95), Tsitsiklis & Van Roy (96), ...
- Limited positive theory or algorithms
  Gordon (96), Tsitsiklis & Van Roy (97), Lagoudakis & Parr (03), Sutton et al. (08, 09), Maei et al. (10), ...

A major open problem for decades.

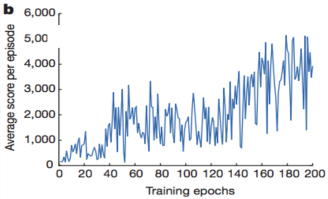
Functional Approximations and Dynamic Programming

By Richard Bellman and Stuart Dreyfus

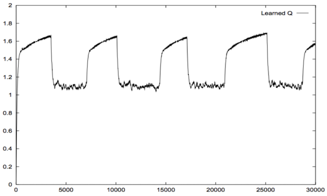
Math. Tables & Other Aids Comp. (1959)

Does It Matter in Practice?

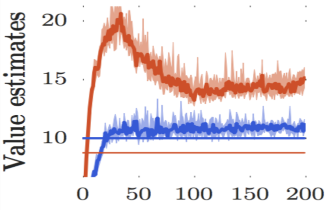
Many empirical successes of (double, dueling) DQN, A3C, ...
in video games, Go, robotics, dialogue management, ...
but often with surprises:



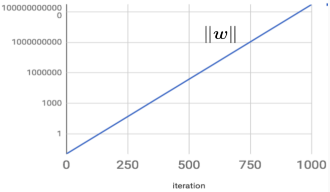
Mnih et al (2015)



Gordon (1996)



van Hasselt et al. (2016)



Baird (1993); Tsitsiklis & Van Roy (1996)

The Deadly Triad (Controlled Case)

Existing RL algorithms risk divergence in the "deadly triad":

- (nonlinear) function approximation
- bootstrapping
- off-policy learning

Outline

- Background
- **Saddle-point Reformulation of Bellman Equation**
- SBEED Learning
- Conclusions

Why Solving Bellman Equation is Hard?

A natural objective function for solving $V = \mathcal{T}V$:

$$\min_V \quad \underbrace{\|V - \mathcal{T}V\|^2}_{\text{“Bellman error/residual”}}$$

$$= \quad \min_V \mathbb{E}_s \left[\left(V(s) - \max_{\mathbf{a}} (R(s,a) + \gamma \mathbb{E}_{\mathbf{s'|s,a}}[\mathbf{V(s')}]) \right)^2 \right]$$

- Difficulty #1: breaks smoothness and continuity
- Difficulty #2: typical SGD gives biased gradient, known as “double sample” issue (Baird 95):

$$\left(\cdots + \gamma \mathbb{E}_{s'|s,a}[V_w(s')] \right)^2 \quad \neq \quad \mathbb{E}_{s'|s,a} \left[(\cdots + \gamma V_w(s'))^2 \right]$$

Addressing Difficulty #1: Nesterov Smoothing

$$V(s) \quad = \quad \max_a \left(R(s,a) + \gamma \mathbb{E}_{s'|s,a}[V(s')] \right)$$

$$\Downarrow$$

$$V(s) \quad = \quad \underbrace{\max_{\pi(\cdot|s)} \sum_a \pi(a|s) \left(R(s,a) + \gamma \mathbb{E}_{s'|s,a}[V(s')] \right) \quad \underbrace{+\lambda H(\pi(\cdot|s))}_{\text{entropic regularization}}}_{:= \mathcal{T}_\lambda V(s)}$$

The smoothed Bellman operator $\mathcal{T}_\lambda$ may be derived differently

Rawlik+ (12), Fox+ (16), Ne+ (17), Nachum+(17), Asadi & Littman (17), ...

Properties of $\mathcal{T}_\lambda$

$$\boxed{\mathcal{T}_\lambda V(s) := \max_{\pi(\cdot|s)} \sum_a \pi(a|s) \left(R(s,a) + \gamma \mathbb{E}_{s'|s,a}[V(s')] \right) + \lambda H(\pi(\cdot|s))}$$

- Still a γ-contraction
- Existence and uniqueness of fixed point V_λ^*
- Controlled bias: $\|V_\lambda^* - V^*\|_\infty = \mathcal{O}(\lambda/(1-\gamma))$
- Temporal consistency (as in PCL of Nachum+ (17))

$$\forall s, a : \quad V_\lambda^*(s) = R(s,a) + \gamma \mathbb{E}_{s'|s,a}[V_\lambda^*(s)] - \lambda \log \pi_\lambda^*(a|s)$$

Addressing Difficulty #2: Legendre-Fenchel Transformation

$$\min_{V} \mathbb{E}_s \left[\left(V(s) - \max_{\mathbf{a}} (R(s,a) + \gamma \mathbb{E}_{\mathbf{s'|s,a}}[\mathbf{V(s')}] \right)^2 \right]$$

$$\Downarrow \quad \text{(by Nesterov smoothing)}$$

$$\min_{V,\pi} \mathbb{E}_{s,a} \left[\left(\underbrace{R(s,a) + \gamma \mathbb{E}_{\mathbf{s'|s,a}}[\mathbf{V(s')}] - \lambda \log \pi(a|s) - V(s)}_{\text{denoted } x_{sa}} \right)^2 \right]$$

$$\Downarrow \quad \text{(L-F transform: } x_{sa}^2 = \max_{y \in \mathbb{R}} (2x_{sa} y - y^2))$$

$$\min_{V,\pi} \max_{\nu \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}} \mathbb{E}_{s,a,s'} \left[(2\nu(s,a) x_{s,a} - \nu(s,a)^2) \right]$$

The last step also applies the interchangeability principle

(Rockafellar & Wets 88; Shapiro & Dentcheva 14; Dai+ 17)

Reformulation of Bellman Equation

We have now turned a fixed point into a saddle point:

$$\min_{V,\pi} \max_{\nu} \mathbb{E}_{s,a,s'} \Big[2\nu(s,a) \Big(R(s,a) + \gamma V(s') - \lambda \log \pi(a|s) - V(s) \Big) - \nu(s,a)^2 \Big]$$

- Well-defined objective without requiring double samples
- May be optimized by gradient methods (SGD/BackProp, ...)
- See paper for a slightly more general version
- Special cases: GTD2, PCL, ...
- Inner maximum achieved when ν equals λ-smoothed Bellman error

Outline

- Background
- Saddle-point Reformulation of Bellman Equation
- **SBEED Learning**
- Conclusions

SBEED: Smoothed Bellman Error EmbeDding

Algorithmic ideas

- Parameterize $(V, \pi; \nu)$ by $(w_V, w_\pi; w_\nu)$
- Stochastic-gradient updates on (w_V, w_π) and ascent on w_ν
  - Two-time-scale updates for primal and dual variables; or
  - Exact maximization if concave in w_ν
- Our implementation uses stochastic mirror descent

Algorithmic advantages

- Agnostic to whether data is on- or off-policy
- May be used in batch (e.g., experience replay) or online mode
- Extensible to multi-step and eligibility traces cases
- Efficiently implemented (only first-order updates needed)

SBEED Convergence (Batch Case)

Define $\bar{\ell}(V, \pi) := \max_\nu L(V, \pi, \nu)$, and assume

- $\nabla \bar{\ell}$ is Lipschitz-continuous
- the stochastic gradient has finite variance
- stepsizes are properly set

Theorem. SBEED solution satisfies $\mathbb{E}[\|\nabla \bar{\ell}(V_{\hat{w}}, \pi_{\hat{w}})\|] \to \mathbf{0}$

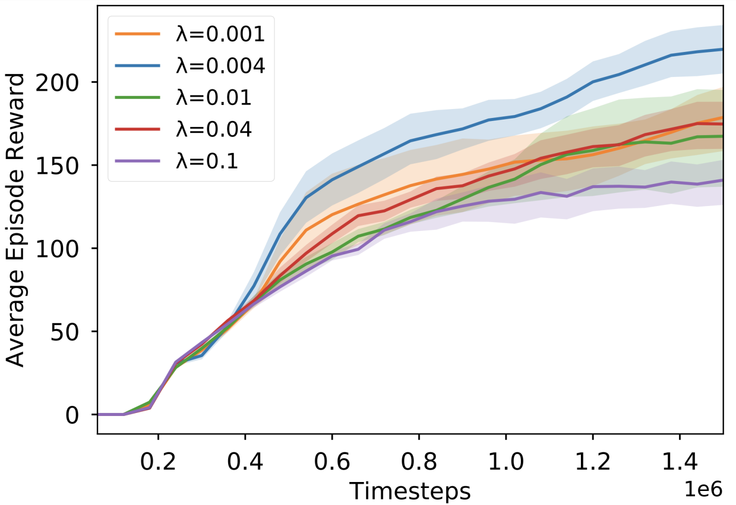
- Building on results of Ghadimi & Lan (13)
- See paper for variants of convergence results ...
- ... and statistical/generalization analyses

Experiments

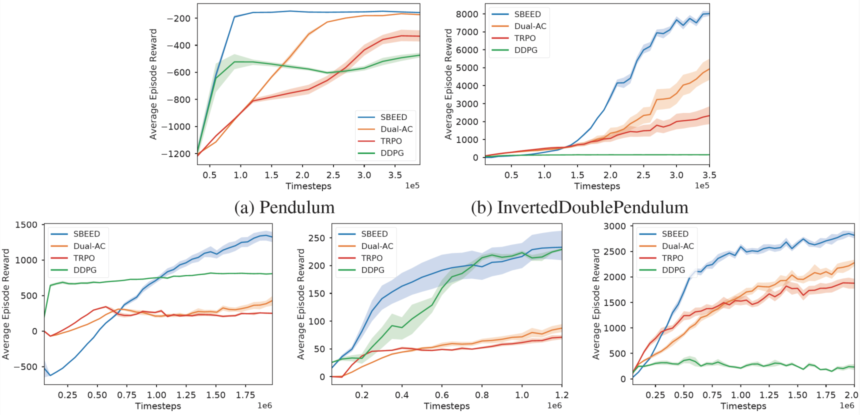
- Use Mujoco on OpenAI as benchmark
- Compare to state-of-the-art baselines:
  - Dual-AC (Dai et al. 18)
  - TRPO (Schulman et al. 15)
  - DDPG (Lillicrap et al. 15)

(from http://www.mujoco.org)

Role of Smoothing Parameter λ



Comparison against Baselines



(a) Pendulum (b) InvertedDoublePendulum

(c) HalfCheetah (d) Swimmer (e) Hopper

Outline

- Background
- Saddle-point Reformulation of Bellman Equation
- SBEED Learning
- **Conclusions**

Conclusions

Contributions

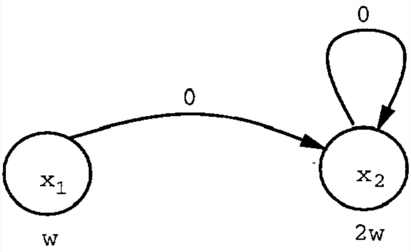
- A saddle-point reformulation of Bellman equation
- New algorithm SBEED with guaranteed convergence
- Promising empirical results on standard benchmark

Further remarks

- Many directions for future research
- Much efforts of finding true gradient RL algorithms
  GTD (Sutton et al. 08), GTD2 (Sutton et al. 09), ...
- Deep connection to optimization
  Mahadevan et al. (14), Macua et al. (15), ...
- Stronger algorithms based on new optimization techniques
  Liu et al. (15, 16), Dai et al. (17, 18), Du et al. (17), Wang (17), Chen et al. (18), ...

APPENDIX

Divergence Example of Tsitsiklis & Van Roy (96)



Starting with $w^{(0)} \neq 0$,
least-squares value iteration diverges when $\gamma > 5/6$,
although V^* may be exactly represented (with $w^* = 0$).

Online SBEED Learning with Experience Replay

Algorithm 1 Online SBEED learning with experience replay

Initialize $w = (w_V, w_\pi, w_\rho)$ and π_b randomly, set ϵ.
for episode $i = 1, \ldots, T$ **do**
  for size $k = 1, \ldots, K$ **do**
    Add new transition (s, a, r, s') into $\mathcal{D}$ by executing behavior policy π_b.
  end for
  for iteration $j = 1, \ldots, N$ **do**
    Update w_ρ^j by solving

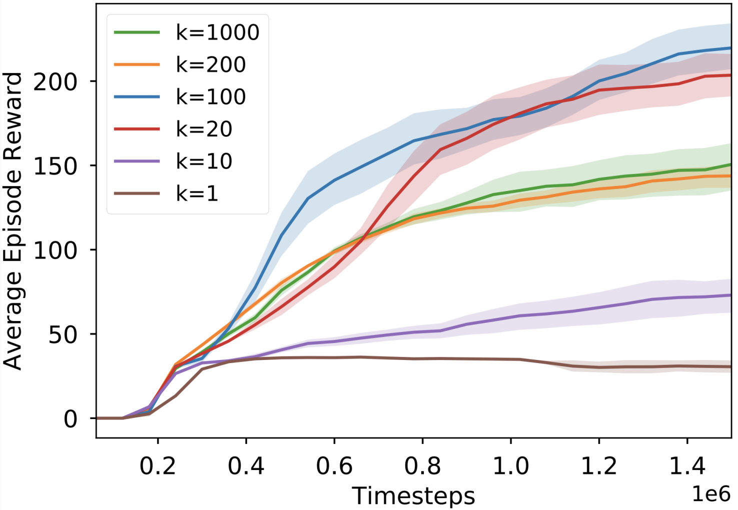
$$\min_{w_\rho} \widehat{\mathbb{E}}_{\{s,a,s'\}\sim\mathcal{D}} \left[\left(\delta(s,a,s') - \rho(s,a) \right)^2 \right].$$

    Decay the stepsize ζ_j in rate $\mathcal{O}(1/j)$.
    Compute the stochastic gradients w.r.t. w_V and w_π as $\widehat{\nabla}_{w_V} \bar{\ell}(V, \pi)$ and $\widehat{\nabla}_{w_\pi} \bar{\ell}(V, \pi)$.
    Update the parameters of primal function by solving the prox-mappings, *i.e.*,

$$\begin{aligned} \text{update } V: &\quad w_V^j = P_{w_V^{j-1}}\left(\zeta_j \widehat{\nabla}_{w_V} \bar{\ell}(V, \pi)\right) \\ \text{update } \pi: &\quad w_\pi^j = P_{w_\pi^{j-1}}\left(\zeta_j \widehat{\nabla}_{w_\pi} \bar{\ell}(V, \pi)\right) \end{aligned}$$

  end for
  Update behavior policy $\pi_b = \pi^N$.
end for

Role of Bootstrapping Distance k



Role of Dual Embedding η