Variance Reduced Policy Evaluation with Smooth Function Approximation

Hoi-To Wai (CUHK), Mingyi Hong (UMN), Zhuoran Yang (Princeton), Zhaoran Wang (Northwestern), Kexin Tang (UMN)

Motivation

- **Policy evaluation** (PE) evaluates the *value function* of average reward at a state, given a policy.
- For large state space, *nonlinear* (and smooth) function approximation is widely used, e.g., neural net.

Aim: Theoretical study of an efficient algorithm for policy evaluation with nonlinear function approx..

Problem Formulation

Discounted MDP: $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$

- $\mathcal{S}$ – state space, $\mathcal{A}$ – action space.
- $\mathcal{P}^a : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ – Markov kernel for state transition under action $a \in \mathcal{A}$.
- $\mathcal{R}(s,a)$ – reward at state s and under action a.
- $\gamma \in (0,1)$ – discount factor.

Policy: π is a conditional probability $\pi(a|s)$ of choosing action a under state s.

Goal: given a policy π, learn the value function

$$V^\pi(s) := \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t,a_t) \Big| \begin{array}{l} s_0 = s, a_t \sim \pi(\cdot|s_t), \\ s_{t+1} \sim \mathcal{P}^{a_t}(s_t, \cdot) \end{array}\Big]$$

PE can be **solved** by

$$\text{(Bellman eq.)} \Longrightarrow \boldsymbol{V}^\pi(s) = \mathcal{T}^\pi \boldsymbol{V}^\pi(s).$$

where for any measurable function f on $\mathcal{S}$,

$$(\mathcal{T}^\pi f)(s) := \mathbb{E}[\mathcal{R}(s,a) + \gamma(\mathcal{P}^a f)(s) | a \sim \pi(\cdot|s)]$$

Challenges:

- The state space $\mathcal{S}$ is **large (can be infinite)**.
- State transition probability is **unknown**.

Remedy: nonlinear function approximation:

- Replace $V^\pi(s)$ by a parameterized function $V_{\boldsymbol{\theta}}(s)$
- E.g., $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$ are the weights of a NN.
- **Objective**: find $\boldsymbol{\theta} \in \Theta$ to minimize

$$J(\boldsymbol{\theta}) := \frac{1}{2}\Big\|\Pi(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(\cdot) - V_{\boldsymbol{\theta}}(\cdot))\Big\|^2_{p^\pi(\cdot)}$$

  $p^\pi(\cdot)$ is stationary distribution of s under π and Π is projection onto the function approximation space.
- **Prior work**: [1] studied a TD learning algo.

Projected Bellman Error Minimization as Primal-dual Optimization

- The function $V_{\boldsymbol{\theta}}$ is smooth w.r.t. $\boldsymbol{\theta}$, with gradient $g_{\boldsymbol{\theta}}(s) := (\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}})(s)$ and Hessian $H_{\boldsymbol{\theta}}(s) := (\nabla^2_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}})(s)$.
- Evaluating unbiased stochastic gradient of $J(\boldsymbol{\theta})$ is hard $\because$ sampling from $p^\pi(\cdot)$ and forming $\boldsymbol{G}_{\boldsymbol{\theta}}^{-1}$.
- Define $\boldsymbol{G}_{\boldsymbol{\theta}} := \mathbb{E}_{s\sim p^\pi(\cdot)}[g_{\boldsymbol{\theta}}(s) g_{\boldsymbol{\theta}}^\top(s)]$, the loss function $J(\boldsymbol{\theta})$ admits a Fenchel's dual reformulation [1]:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{s\sim p^\pi(\cdot)}\big[(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)) g_{\boldsymbol{\theta}}(s)^\top\big]\, \boldsymbol{G}_{\boldsymbol{\theta}}^{-1}\, \mathbb{E}_{s\sim p^\pi(\cdot)}\big[(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)) g_{\boldsymbol{\theta}}(s)\big] = \frac{1}{2}\Big\|\mathbb{E}_{s\sim p^\pi(\cdot)}\big[(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)) g_{\boldsymbol{\theta}}(s$$

$$= \max_{\boldsymbol{w}\in\mathbb{R}^d} \Big(-\frac{1}{2}\mathbb{E}_{s\sim p^\pi(\cdot)}\big[(\boldsymbol{w}^\top g_{\boldsymbol{\theta}}(s))^2\big] + \big\langle \boldsymbol{w}, \mathbb{E}_{s\sim p^\pi(\cdot)}\big[(\mathcal{T}^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)) g_{\boldsymbol{\theta}}(s)\big]\big\rangle\Big)$$

Batch RL setting – observe a trajectory of state-action pairs $\{s_1, a_1, s_2, a_2, ..., s_m, a_m, s_{m+1}\}$ generated from π,

$$\min_{\boldsymbol{\theta}\in\Theta} J(\boldsymbol{\theta}) \overset{\text{approx. by}}{\Longrightarrow} \min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{m}\sum_{t=1}^m \mathcal{L}_t(\boldsymbol{\theta}, \boldsymbol{w}) \ \text{ w/ } \ \mathcal{L}_t(\boldsymbol{\theta}, \boldsymbol{w}) = \langle \boldsymbol{w}, g_{\boldsymbol{\theta}}(s_t)(\mathcal{R}(s_t,a_t) + \gamma V_{\boldsymbol{\theta}}(s_{t+1}) - V_{\boldsymbol{\theta}}(s_t))\rangle - \frac{(\boldsymbol{w}^\top g_{\boldsymbol{\theta}}(s_t))^2}{2}$$

- If $\boldsymbol{G}_{\boldsymbol{\theta}}$ = positive definite, inner max. is strongly concave w.r.t. $\boldsymbol{w}$; yet outer min. w.r.t. $\boldsymbol{\theta}$ is *non-convex*.
- A finite-sum, one-sided non-convex primal-dual opt. $\Rightarrow$ natural algo = primal dual gradient descent/ascent.

Nonconvex Primal-Dual Gradient with Variance Reduction (nPD-VR) Algorithm

- Directly optimizing the finite-sum problem has high complexity $\Rightarrow$ SGD is fast but slow convergence...
- **Philosophy:** balance between complexity and speed of convergence $\Rightarrow$ variance reduction via SAGA [2].

for $k \geq 1$ **do**

Select $i_k, j_k \in \{1, ..., m\}$ uniformly and independently.

Primal-dual gradient update through

$$\boldsymbol{\theta}^{(k+1)} = \mathcal{P}_\Theta\Big\{\boldsymbol{\theta}^{(k)} - \beta\Big(\mathsf{G}_{\boldsymbol{\theta}}^{(k)} + \big(\nabla_{\boldsymbol{\theta}}\mathcal{L}_{i_k}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}_{i_k}(\boldsymbol{\theta}_{i_k}^{(k)}, \boldsymbol{w}_{i_k}^{(k)})\big)\Big)\Big\}$$

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \alpha\Big(\mathsf{G}_{\boldsymbol{w}}^{(k)} + \big(\nabla_{\boldsymbol{w}}\mathcal{L}_{i_k}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)}) - \nabla_{\boldsymbol{w}}\mathcal{L}_{i_k}(\boldsymbol{\theta}_{i_k}^{(k)}, \boldsymbol{w}_{i_k}^{(k)})\big)\Big).$$

Update **stored variables** as:

$$\boldsymbol{\theta}_i^{(k+1)} = \begin{cases} \boldsymbol{\theta}^{(k)} & \text{if } i = j_k \\ \boldsymbol{\theta}_i^{(k)} & \text{if } i \neq j_k \end{cases}, \quad \boldsymbol{w}_i^{(k+1)} = \begin{cases} \boldsymbol{w}^{(k)} & \text{if } i = j_k \\ \boldsymbol{w}_i^{(k)} & \text{if } i \neq j_k \end{cases}$$

$$\mathsf{G}_{\boldsymbol{\theta}}^{(k+1)} = \mathsf{G}_{\boldsymbol{\theta}}^{(k)} + \frac{1}{m}\big(\nabla_{\boldsymbol{\theta}}\mathcal{L}_{j_k}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}_{j_k}(\boldsymbol{\theta}_{j_k}^{(k)}, \boldsymbol{w}_{j_k}^{(k)})\big),$$

$$\mathsf{G}_{\boldsymbol{w}}^{(k+1)} = \mathsf{G}_{\boldsymbol{w}}^{(k)} + \frac{1}{m}\big(\nabla_{\boldsymbol{w}}\mathcal{L}_{j_k}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)}) - \nabla_{\boldsymbol{w}}\mathcal{L}_{j_k}(\boldsymbol{\theta}_{j_k}^{(k)}, \boldsymbol{w}_{j_k}^{(k)})\big),$$

Primal-dual SAGA —

- A primal-dual version of non-convex SAGA in [2].
- Update w/ indices i_k, j_k to ensure unbiasedness.
- $\mathcal{O}(d^2)$ FLOPS per iteration (reduced to $\mathcal{O}(d)$ w/ approx.)

Challenges of analysis —

- One-sided non-convexity.
- Algorithm is non-monotone.

Assumptions —

- Strong concavity for $\mathcal{L}$ w.r.t. $\boldsymbol{w}$.
- Lipschitz cts. gradient for $\mathcal{L}$.
- $(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)})_{k=1}^K$ = bounded.

Theorem 1. Choosing step sizes $\beta, \alpha = \Theta(1/m)$. Let $\tilde{K}$ be uniformly picked from $\{1, ..., K\}$. It holds that

$$\mathbb{E}\Big[\frac{1}{\beta^2}\|\bar{\boldsymbol{\theta}}^{(\tilde{K})} - \boldsymbol{\theta}^{(\tilde{K})}\|^2 + \|\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{\theta}^{(\tilde{K})}, \boldsymbol{w}^{(\tilde{K})})\|^2\Big] \leq \frac{F^{(K)} + \frac{4}{\mu}\Big(3 + 2m\big(2L_{\boldsymbol{w}}^2\alpha + L_{\boldsymbol{\theta}}^2\beta\big)\Big)\|\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{\theta}^{(0)}, \boldsymbol{w}^{(0)})\|^2}{K\min\{\alpha, \frac{\beta}{4}\}}$$

- Left hand side is a measure of primal-dual stationarity $\Rightarrow$ convergence rate is roughly $\mathcal{O}(m/K)$.
- Caveat: bounded iterate assumption can be hard to verify, in practice we project $\boldsymbol{w}$ to a bounded set.

Main Steps of Proof

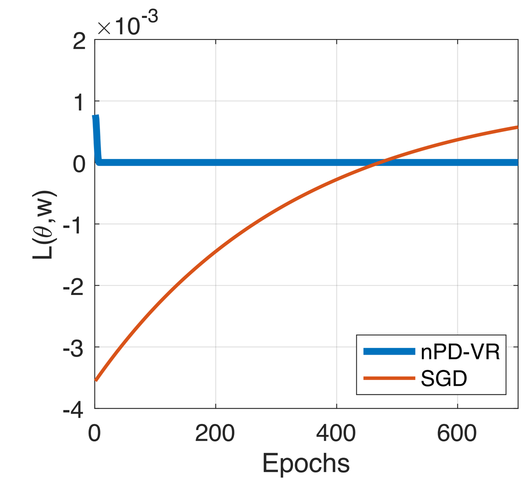
- Bound primal-dual updates' progress on the objective value $\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)})$.
- By carefully controlling the step size, we show

$$\begin{aligned} \Omega(\min\{\alpha,\beta\}) \textstyle\sum_{k=0}^{K-1}\mathbb{E}\big[\mathcal{G}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)})\big] &\leq \mathcal{O}(\alpha) \textstyle\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{w}^{(k)})\|^2] \\ &\quad + \mathcal{O}(m - \tfrac{1}{\beta}) \textstyle\sum_{k=0}^{K-1}\mathbb{E}[\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|^2]. \end{aligned} \quad \text{(A)}$$

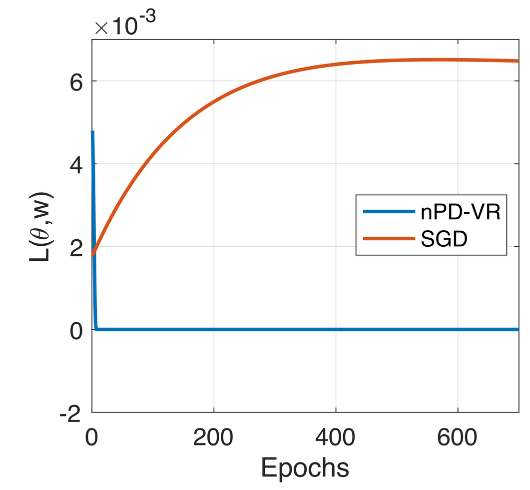
- Involves **new** technique in controlling the error due to SAGA.
- Green term $\leq \sum_{k=0}^{K-1}\mathbb{E}[\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|^2]$.
- Selecting the right step size ensures the RHS of (A) is $\mathcal{O}(1)$.
- Using $\tilde{K} \sim \mathcal{U}\{1, ..., K\}$ finishes the proof.

Preliminary Experiments

- **Setting**: `mountaincar` dataset w/ $m = 5000$.
- Nonlinear function $V_{\boldsymbol{\theta}}(\cdot)$ is parameterized as 2-layer Neural network with n neurons.
- Set constraints as $\Theta = [0,1]^n$ and $\boldsymbol{w} \in [0,100]^n$.
- Step sizes are $\alpha = 10^{-4}$, $\beta = 10^{-8}$.



$(n = 50 \text{ neurons})$



$(n = 100 \text{ neurons})$

- Compared to plain SGD, nPD-VR converges to a stationary point with less no. of epochs.

Future work — mini-batch design to speed up convergence, improve analysis with projection of $\boldsymbol{w}$, etc.

References.

1. S. Bhatnagar, et al. Convergent temporal-difference learning with arbitrary smooth function approximation. NeurIPS 2009.
2. S. J. Reddi, et al. Proximal stochastic methods for nonsmooth non-convex finite-sum optimization. NeurIPS, 2016.

NeurIPS 2019, Vancouver, Canada. Contact E-mail: `htwai@se.cuhk.edu.hk`

Acknowledgement: CUHK Direct Grant #4055113 (HT), NSF-CMMI 1727757, CIF-1910385 (MH), AFORSR 19RT0424 (MH).