Sample Complexity of Reinforcement Learning with a Generative Oracle

Ziniu Li
ziniuli@link.cuhk.edu.cn

The Chinese University of Hong Kong, Shenzhen, Shenzhen, China

Oct 8, 2020

Mainly based on the note:
http://www.liziniu.org/docs/rl-generative-model.pdf

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

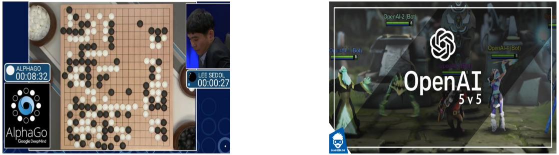
Model-based Methods

- Model-based Value Iteration

Summary

Emerging Applications with Reinforcement Learning

- Recently, there are successful applications with (deep) reinforcement learning (RL).



Figures from Internet.

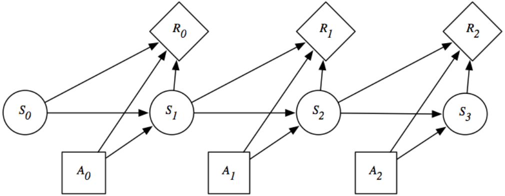
- Existing methods might not be optimal due to lack of theory and full of bag of tricks.
- To design more effective methods, we need the mathematical framework of Markov Decision Process [Puterman, 1994, Sutton and Barto, 2018].

Markov Decision Process

- Consider an infinite-horizon Markov Decision Process $\mathcal{M}^* = (\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$ [Puterman, 1994, Sutton and Barto, 2018].
  - $\mathcal{S}$ and $\mathcal{A}$ are the (finite) state and action space, respectively.
  - P determines the transition probability of s_{t+1} conditioned on s_t and a_t.
  - R is the (bounded) reward function, which assigns a reward $r(s,a)$ for state-action pair (s,a).
  - $\gamma \in [0,1)$ is a discount factor, balancing the importance of future rewards.
  - d_0 specifies the initial state distribution.

Markov Decision Process

- The decision process is characterized as follows:
  - At the beginning of the epoch, the environment resets to some initial state s_0 according to d_0;
  - The agent observes the state s_0 and selects an action a_0 to perform;
  - The environment transits to s_1 according to P and sends a reward signal r_0 to the agent.
  - This process repeats until some terminal signal is released, after which the environment resets to some initial state again.



Markov Decision Process

- The above action selection procedure can be described as a policy $\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$, which maps the state space to a probability simplex over the action space.
- The goal of an intelligent agent is to maximize its payoff by searching the optimal policy π^* with maximal cumulative rewards.

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right].$$

- Though the above decision-making procedure seems endless, the effective planning horizon is $1/(1-\gamma)$.

$$\mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \leq \frac{R_{\max}}{1-\gamma},$$

where $R_{\max}$ is the maximal reward, which is assumed to be 1 without loss of generality.

Value Function

- The (state) value function (or V-function) for an infinite-horizon MDP is defined as:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) | s_0 = s, a_k \sim \pi(\cdot|s), k \geq 0\right].$$

- Similarly, the (state-action) value function (or Q-function) is defined as:

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) | s_0 = s, a_0 = a, a_{k+1} \sim \pi(\cdot|s), k \geq 0\right]$$

- The policy value is defined as the expected long-term return:

$$V(\pi) = \mathbb{E}_{s_0 \sim \rho(s)}\left[V^{\pi}(s_0)\right].$$

Bellman Optimality Equation

- The Bellman Optimality Equation for V-function and Q-function is defined as:

$$\begin{cases} V(s) &= \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)} \left[V(s')\right]\right] \qquad (V\text{-function}) \\ Q(s,a) &= r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)} \left[\max_{a'\in\mathcal{A}} Q(s',a')\right] \qquad (Q\text{-function}) \end{cases} \tag{1}$$

- Define the optimal (state/state-action) value function as:

$$V^*(s) = \max_{\pi} V^{\pi}(s), \quad Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \qquad \forall (s,a) \in \mathcal{S} \times \mathcal{A}.$$

Bellman Operator for V-function

- The (population-based) Bellman operator $\mathcal{T}$ for V-function is a mapping from $\mathbb{R}^{|\mathcal{S}|}$ to itself:

$$\mathcal{T}(V)(s) \equiv \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[V(s') \right] \right].$$

- It can be proved V^* is the unique solution to Equation (1) [Puterman, 1994].

$$V^* = \mathcal{T}(V^*).$$

- Thus, repeatedly applying Bellman operator from any point converges to the optimal state value function.

Bellman Operator for Q-function

- Similarly, the (population-based) Bellman operator $\mathcal{T}$ for Q-function is a mapping from $\mathbb{R}^{|\mathcal{S}|\times\mathcal{A}}$ to itself:

$$\mathcal{T}(Q)(s,a) \equiv r(s,a) + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\max_{a'\in\mathcal{A}} Q(s',a')\right].$$

- Similarly, Q^* is the unique solution to Equation (1) [Puterman, 1994].

$$Q^* = \mathcal{T}(Q^*).$$

- Again, repeatedly applying Bellman operator from any point converges to the optimal state-action value function.

Properties of Bellman Operator

- (γ-contractive) For any two value function V_1 and V_2, we have

$$\|\mathcal{T}(V_1) - \mathcal{T}(V_2)\|_\infty \leq \gamma \|V_1 - V_2\|_\infty .$$

  - Corollary: $||\mathcal{T}(V) - V^*||_\infty \leq \gamma ||V - V^*||_\infty$.

- (Order-reserving) For any two V_1 and V_2 satisfying that $V_1 \leq V_2$ ($\leq$ holds elementwise), we have

$$\mathcal{T}(V_1) \leq \mathcal{T}(V_2).$$

- The above properties also hold for Q-function.

Methods to Markov Decision Process

- Contraction-based Method
  - Value-based: (Q or V) value iteration.
  - Policy-based: policy iteration; policy gradient.
- Not-Contraction-based method
  - Linear programming [Puterman, 1994] (notes on this will be released later).

Main Algorithm of Value Iteration

Algorithm 1 Value Iteration

```
Input: initial value $V_0$ and iteration number $L$.
Initialize an auxiliary variable $Z \in \mathbb{R}^{|\mathcal{S}|}$.
for $\ell = 1, 2, \cdots, L$ do
    for each state $s \in \mathcal{S}$ do
        % Performing population-based Bellman update.
        $Z(s) = \max_{a \in \mathcal{A}} \left[ r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[ V_{\ell-1}(s') \right] \right]$.
    end for
    Set $V_\ell = Z$.
end for
Output: $V_L$.
```

Theoretical Guarantee for Value Iteration

Theorem 1 (Linear Convergence of Value Iteration).

With the parameter

$$L = \left\lceil \frac{1}{1-\gamma} \log \frac{1}{\epsilon} \right\rceil,$$

Value Iteration (see Algorithm 1) can find a sub-optimal value function V_L *such that* $||V_L - V^*||_\infty \leq \epsilon$ *from any initial solution* V_0*, where* $\epsilon \in (0, \frac{1}{1-\gamma}]$ *is the error tolerance.*

Task of Reinforcement Learning

- Setting of Reinforcement Learning [Sutton and Barto, 2018]:
  - Transition probability P is unknown.
  - Reward function R is unknown (option).
  - Interaction with the environment is allowed.
- Goal: quickly find an ϵ-optimal policy π.
  - This can also be achieved by learning an $\epsilon/(1-\gamma)$-optimal Q-function, upon which we derive a greedy policy $\pi(s) = \arg\max_a Q(s, a)$ [Bertsekas and Tsitsiklis, 1996].

$$0 \leq V(\pi^*) - V(\pi) \leq \epsilon.$$

Setting of RL: Online Learning

- In addition to previous conditions, the environment/simulator can only start from some initial states.
- Interactions come from in the way of stream-data, a.k.a., online learning.
- The leaner needs to balance the trade-off of exploration and exploitation.
- Not the focus of this presentation.

Setting of RL: Generative Oracle

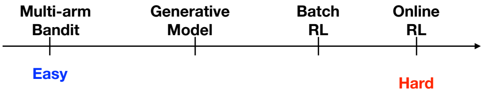
- Generative Oracle $\mathcal{M}$: we can directly reset it to any state s_t, after which we can take an action a_t and observe the next state $s_{t+1} \sim p(\cdot|s_t, a_t)$ and the reward $r(s_t, a_t)$.
  - Compared to the pure MDP problem, we still do not known P in advance.
  - Compared to the online RL problem, we can go to any s_t without the planning from an initial state s_0.
  - In particular, we have access to the whole state space and action space (i.e., no exploration issue).
- Example: a perfect simulator (e.g., some video game simulators), where we can load (reset) the state s_t from RAM.
- The main focus of this presentation.

Setting of RL: Offline Learning

- The learner cannot interact with the environment, but is provided with some fixed dataset.
- The learner needs to make safe improvement from this insufficient dataset.
- Not the focus of this presentation.

Setting Comparison

- Difficulty comparison with different settings:



Lower Bound with a Generative Oracle

Definition 2 ((ϵ, δ)-correct algorithm).

Let V be the output of some RL algorithm $\mathbb{A}$. We say that $\mathbb{A}$ is (ϵ, δ)-correct on the class of MDPs $\mathbb{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_m\}$ if $||V^*_{\mathcal{M}} - V||_\infty \leq \epsilon$ with probability at least $1 - \delta$ for each $\mathcal{M} \in \mathbb{M}$.

Theorem 3 (Lower bound with generative oracle [Azar et al., 2013]).

There exist some constants $\epsilon_0, \delta_0, c_1, c_2$ and a class MDPs $\mathbb{M}$, such that for all $\epsilon \in (0, \epsilon_0)$, $\delta \in (0, \delta_0)$, and every (ϵ, δ)-correct RL algorithm on the class of MDPs $\mathbb{M}$, the total number of state-transition samples needs to be at least

$$T = \left\lceil \frac{|\mathcal{S}| \times |\mathcal{A}|}{c_1 \epsilon^2 (1-\gamma)^3} \log\left(\frac{|\mathcal{S}| \times |\mathcal{A}|}{c_2 \delta}\right) \right\rceil .$$

Comment on Lower Bound

- Define $T_{\mathcal{M}}(\mathbb{A})$ as the number of samples of algorithm $\mathbb{A}$ to get an ϵ-accurate solution on MDP $\mathcal{M}$ with probability at least $1-\delta$.
- Understanding the lower bound and upper bound:

$$\underbrace{T_{\mathcal{M}}(\mathbb{A})}_{\text{(actual performance)}} \leq \underbrace{\sup_{\mathcal{M}} T_{\mathcal{M}}(\mathbb{A})}_{\text{(upper bound)}},$$

$$\underbrace{\inf_{\mathbb{A}} \sup_{\mathcal{M}} T_{\mathcal{M}}(\mathbb{A})}_{\text{(lower bound)}} \leq \underbrace{\sup_{\mathcal{M}} T_{\mathcal{M}}(\mathbb{A})}_{\text{(upper bound)}}.$$

- An algorithm $\mathbb{A}$ is sailed to be minimax-optimal if its upper bound matches the lower bound (constant and logarithmic terms can be ignored).

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Phased Value Iteration

- A simple method with a generative oracle is to replace the population-based Bellman operator $\mathcal{T}$ with sample-average-approximation, see Algorithm 2 [Kearns and Singh, 1999].

Algorithm 2 Phased Value Iteration

```
Input: initial value $V_0$, iteration number $L$, and sample size $n$.
Initialize $\hat{V}_0 = V_0$ and a policy $\hat{\pi}_0 \in \mathbb{R}^{|\mathcal{S}|}$.
Initialize an auxillary variable $Z \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$.
for $\ell = 1, 2, \cdots, L$ do
    for each state $s \in \mathcal{S}$ do
        for each any $a \in \mathcal{A}$ do
            % Sample-average-approximation to $\mathcal{T}$.
            Sample $n$ next states $\{s_i'\}_{i=1}^n$ by calling $\mathcal{M}$.
            Set $Z(s,a) = r(s,a) + \gamma \sum_{i=1}^n \frac{1}{n} \hat{V}_{\ell-1}(s_i')$.
        end for
        Set $\hat{V}_\ell(s) = \max_{a \in \mathcal{A}} Z(s,a)$ and $\hat{\pi}_\ell(s) = \arg\max_{a \in \mathcal{A}} Z(s,a)$.
    end for
end for
Output: $(\hat{V}_L, \hat{\pi}_L)$.
```

Theoretical Guarantee for Phased Value Iteration

Theorem 4 (Sample Complexity of Phased Value Iteration).

Given a generative oracle $\mathcal{M}$, with the parameters:

$$L = \left\lceil \frac{1}{1-\gamma} \log \frac{2}{(1-\gamma)\epsilon} \right\rceil, \quad n = \frac{4}{\epsilon^2 (1-\gamma)^4} \log \left(\frac{2 \times |\mathcal{S}| \times |\mathcal{A}| \times T}{\delta} \right), \quad V_0 = 0,$$

Phased Value Iteration (see Algorithm 2) ensures that $||V^* - \hat{V}_L||_\infty \leq \epsilon$ with probability at least $1-\delta$, and the number of total samples used is

$$\mathcal{O}\left(\frac{|\mathcal{S}| \times |\mathcal{A}|}{\epsilon^2 (1-\gamma)^5} \log \left(\frac{1}{\delta} \right) \right).$$

Proof Idea of Theorem 4

- Suppose we can choose a large sample number n to ensure that sampling-based $\widehat{\mathcal{T}}$ is accurate such that for any state s, we have

$$\left|\sum_{i=1}^{n} \frac{1}{n} \hat{V}_t(s_i') - \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[\hat{V}_t(s')\right]\right| \leq \epsilon_n.$$

- Based on the assumption, we can prove that the “flaw” of $\widehat{\mathcal{T}}$ is:

$$\left\|V_\ell - \hat{V}_\ell\right\|_\infty \leq \gamma \left\|V_{\ell-1} - \hat{V}_{\ell-1}\right\|_\infty + \gamma\epsilon_n \overset{V_{\ell-1}=\hat{V}_{\ell-1}}{=\!=} \gamma\epsilon_n.$$

  - V_ℓ: the ℓ-th iterator of Value Iteration.
  - $\hat{V}_\ell$: the ℓ-th iterator of Phased Value Iteration.

For any state $s \in \mathcal{S}$,

$$\begin{aligned}
&\left|V_\ell(s) - \hat{V}_\ell(s)\right| \\
&= \left|\max_{a\in\mathcal{A}} \left\{r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[V_{\ell-1}\left(s'\right)\right]\right\} - \max_{a\in\mathcal{A}}\left\{r(s,a) + \gamma \mathbb{E}_{s'\sim \hat{p}(\cdot|s,a)}\left[\hat{V}_{\ell-1}\left(s'\right)\right]\right\}\right| \\
&\overset{(i)}{\leq} \max_{a\in\mathcal{A}}\left|\left\{r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[V_{\ell-1}\left(s'\right)\right]\right\} - \left\{r(s,a) + \gamma \mathbb{E}_{s'\sim \hat{p}(\cdot|s,a)}\left[\hat{V}_{\ell-1}\left(s'\right)\right]\right\}\right| \\
&= \gamma \max_{a\in\mathcal{A}}\left|\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[V_{\ell-1}\left(s'\right)\right] - \mathbb{E}_{s'\sim \hat{p}(\cdot|s,a)}\left[\hat{V}_{\ell-1}\left(s'\right)\right]\right| \\
&\overset{(ii)}{\leq} \gamma \max_{a\in\mathcal{A}}\left|\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[V_{\ell-1}\left(s'\right)\right] - \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[\hat{V}_{\ell-1}\left(s'\right)\right]\right| + \gamma\epsilon_n \\
&\leq \gamma \max_{s'\in\mathcal{S}}\left|V_{\ell-1}\left(s'\right) - \hat{V}_{\ell-1}\left(s'\right)\right| + \gamma\epsilon_n,
\end{aligned} \tag{2}$$

- Recall that the optimality gap shrinks with a linear speed:

$$\left\|V_\ell - \hat{V}_\ell\right\|_\infty \leq \gamma \left\|V_{\ell-1} - \hat{V}_{\ell-1}\right\|_\infty + \gamma\epsilon_n. \tag{3}$$

- Repeatedly applying the above inequality, we get the optimality gap:

$$\begin{aligned}\left\|V_\ell - \hat{V}_\ell\right\|_\infty \leq \gamma^t ||V_0 - \hat{V}_0||_\infty + \sum_{i=1}^{\ell} \gamma^i \epsilon_n \\ \leq \frac{1}{1-\gamma}\epsilon_n,\end{aligned}$$

where we assume that $V_0 = \hat{V}_0$ and ϵ_n is an iteration-independent term.

- By triangle inequality, suppose the sampling-based Bellman update is accurate such that

  $$\frac{1}{1-\gamma}\epsilon_n \leq \frac{1}{2}\epsilon \quad \Longrightarrow \quad \epsilon_n \leq \frac{1-\gamma}{2}\epsilon,$$

  then we have:

  $$\left\|\hat{V}_\ell - V^*\right\|_\infty \leq \underbrace{\left\|V_\ell - \hat{V}_\ell\right\|_\infty}_{\frac{1}{2}\epsilon} + \underbrace{\|V_\ell - V^*\|_\infty}_{\frac{1}{2}\epsilon} \leq \epsilon.$$

- By Hoeffding's inequality, we require that the sample size $n \sim \mathcal{O}(\frac{1}{(1-\gamma)^4\epsilon^2})$.
- It remains to note that the total iteration number $L \sim \mathcal{O}(\frac{1}{1-\gamma})$.

Comment on Phased Value Iteration

- (Uniform Convergence) Phased Value Iteration requires fresh data to update each value function iterator.
  - The accuracy is uniform over iterations, which has the same order with the final accuracy.
- (Error Bound of Induced Policy) The induced greedy policy $\widehat{\pi}_L$ suffers much from the inaccuracy of $\hat{V}_L$, that is,

$$\left\| V^{\widehat{\pi}_L} - V^{\pi^*} \right\|_\infty \le \mathcal{O}\left(\frac{1}{\epsilon^2 (1-\gamma)^7}\right).$$

  - (Lemma [Bertsekas and Tsitsiklis, 1996]) For any value function $\hat{V}$ such that $||\hat{V} - V^*||_\infty \le \epsilon$, suppose that $\hat{\pi}$ is the induced greedy policy by $\hat{V}$, then

$$||V^{\hat{\pi}} - V^{\pi^*}||_\infty \le \frac{2\gamma}{1-\gamma}\epsilon.$$

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Phased Value Iteration: Stochastic Approximation

- Phased Value Iteration (actually all methods with a generative oracle) is a stochastic approximation (SA) method to solve the Bellman equation.

$$V = \mathcal{T}(V) \quad \Longrightarrow \quad V_{t+1} = 0 \cdot V_t + \widehat{\mathcal{T}}(V_t).$$

- By stochastic approximation, there is always sampling-noise in the update, which precludes convergence to the fixed point.
- The same issue also holds for the stochastic gradient descent (SGD) [Johnson and Zhang, 2013].

$$\nabla F(x) = 0 \quad \Longrightarrow \quad x_{t+1} = x_t - \eta_t \nabla f_i(x_t).$$

Improve Phased Value Iteration with SA

- Technically, reducing the variance of noise is to control the estimate of value range when applying Hoeffding's inequality.

$$\mathbb{E}[X^2] = \mathrm{Var}[X] + \mathbb{E}^2[X].$$

- Though naively annealing the stepsize could reduce the variance, which is not the optimal method (this also is true for SGD [Johnson and Zhang, 2013]).

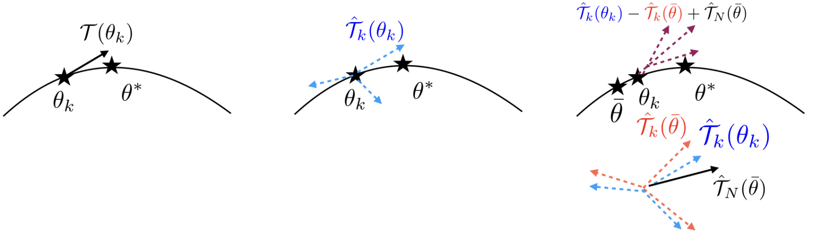
Variance-reduced Value Iteration

- We consider the control variate method:

$$\begin{aligned} V &:= \widetilde{\mathcal{T}}(\tilde{V}) + \widehat{\mathcal{T}}(V) - \widehat{\mathcal{T}}(\tilde{V}) && && (4) \\ &\Longrightarrow \widetilde{\mathcal{T}}(\tilde{V}) + \cancel{\widehat{\mathcal{T}}(V) - \widehat{\mathcal{T}}(\tilde{V})} && (V \approx \tilde{V}) && (5) \\ &\Longrightarrow \mathcal{T}(V) && (\widetilde{\mathcal{T}} \approx \mathcal{T}) && (6) \end{aligned}$$

- We introduce an auxillary iterator $\tilde{V}$ and (sampling-based) Bellman operator $\widetilde{\mathcal{T}}$ to eliminate the sampling noise.
  - $\tilde{V}$ could be the previous iterator.
  - Each iteration, using the samples to update both V and $\tilde{V}$.
  - $\widetilde{\mathcal{T}}(\tilde{V})$ does not change over iterations within the loop.

Illustration of Variance-reduced Value Iteration


$\mathcal{T}(\theta_k)$
θ_k
θ^*
$\hat{\mathcal{T}}_k(\theta_k)$
θ_k
θ^*
$\hat{\mathcal{T}}_k(\theta_k) - \hat{\mathcal{T}}_k(\bar{\theta}) + \hat{\mathcal{T}}_N(\bar{\theta})$
$\bar{\theta}$
θ_k
θ^*
$\hat{\mathcal{T}}_k(\bar{\theta})$
$\hat{\mathcal{T}}_k(\theta_k)$
$\hat{\mathcal{T}}_N(\bar{\theta})$

Expected Bellman Update

Q-learning

Variance-Reduced Q-learning

Why Variance-reduced?

- Rearrange the estimator formula:

$$
\begin{aligned}
V &:= \widetilde{\mathcal{T}}(\tilde{V}) + \widehat{\mathcal{T}}(V) - \widehat{\mathcal{T}}(\tilde{V}) \\
&:= \underbrace{\widetilde{\mathcal{T}}(\tilde{V})}_{\text{one-pass}} + \underbrace{\widehat{\mathcal{T}}(V - \tilde{V})}_{\text{many-pass}}.
\end{aligned}
$$

- The first term only requires samples before the iteration, whose value range estimate cannot be improved.
- The second term requires samples within the iteration, whose value range estimate is reduced (since $V \approx \tilde{V}$).

Main-Algorithm of Variance-reduced Value Iteration

Algorithm 3 Sublinear Randomized Value Iteration: `SublinearRandomizedVI`(ϵ, δ)

```
Input: desired precision $\epsilon$ and failure probability $\delta \in (0,1)$.
Set $K = \left\lceil \log_2 \left( \frac{1}{\epsilon(1-\gamma)} \right) \right\rceil$, and $L = \left\lceil \frac{1}{1-\gamma} \log \left( \frac{4}{1-\gamma} \right) \right\rceil$
Set $V_0 = \vec{0}$ and $\epsilon_0 = \frac{1}{1-\gamma}$.
for each iteration $k = 1, 2, \cdots, K$ do
    Set $\epsilon_k = \frac{1}{2}\epsilon_{k-1} = \frac{1}{2^k(1-\gamma)}$.    % Iteratively shrink the estimate range
    $(V_k, \pi_k) = \mathtt{SampledRandomizedVI}(V_{k-1}, L, (1-\gamma)\epsilon_k/(4\gamma), \delta/K)$.    % Variance-reduced update
end for
Output: $(V_K, \pi_K)$.
```

Sub-Algorithm of Variance-reduced Value Iteration

Algorithm 4 Sampled Randomized Value Iteration: $\texttt{SampledRandomizedVI}(V_0, L, \epsilon, \delta)$

```
Input: initial value $V_0$ and number of iterations $L > 0$
Input: target accuracy $\epsilon > 0$ and failure probability $\delta \in (0,1)$
% Estimate the control variate
Sample $n$ samples to obtain approximate offsets: $\mathcal{X} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ with $|\mathcal{X}(s,a) - \mathbb{E}_{s'\sim p(\cdot|s,a)}[V_0(s')]| \leq \epsilon$ for all $(s,a)$.
    $$n = \left\lceil \frac{2||V_0||_\infty^2}{\epsilon^2} \log\left(\frac{2L}{\delta}\right) \right\rceil .$$
% Single Epoch of Variance-reduced update
for each round $\ell = 1, 2, \cdots L$ do
    $(V_\ell, \pi_\ell) = \texttt{ApxVal}(V_{\ell-1}, V_0, \mathcal{X}, \epsilon, \delta/(2L))$.
end for
Output: $(V_L, \pi_L)$.
```

Sub-Algorithm of Variance-reduced Value Iteration

Algorithm 5 Approximate Value Operator: $\texttt{ApxVal}(V, \tilde{V}, \mathcal{X}, \epsilon, \delta)$

```
Input: current value $V \in \mathbb{R}^{|\mathcal{S}|}$, and reference-point $\tilde{V} \in \mathbb{R}^{|\mathcal{S}|}$.
Input: precomputed offset $\mathcal{X} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ such that $|\mathcal{X}(s,a) - \mathbb{E}_{s' \sim p(\cdot|s,a)}[\tilde{V}(s')]| \leq \epsilon$ for all $(s,a)$.
Input: desired accuracy $\epsilon \in (0,1)$ and failure probability $\delta \in (0,1)$.
Set $n = \left\lceil \frac{2||V-\tilde{V}||_\infty^2}{\epsilon^2} \log\left(\frac{2}{\delta}\right) \right\rceil$.
Initialize variables $Z_t \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$, $\bar{V} \in \mathbb{R}^{|\mathcal{S}|}$ and $\pi \in \mathbb{R}^{|\mathcal{S}|}$.
for each state $s \in \mathcal{S}$ do
    for each action $a \in \mathcal{A}$ do
        Sample $n$ next states $\{s_i'\}_{i=1}^n$ by calling $\mathcal{M}$.
        % Variance-reduced update, see Equation (4)
        Set $Z(s,a) = r(s,a) + \gamma \left( \mathcal{X}(s,a) + \sum_{i=1}^n \frac{1}{n}[V(s_i') - \tilde{V}(s_i')] \right)$.
    end for
    Set $\bar{V}(s) = \max_{a \in \mathcal{A}} Z(s,a)$, and $\pi(s) = \arg\max_{a \in \mathcal{A}} Z(s,a)$.
end for
Output: $(\bar{V}, \pi)$.
```

Theoretical Guarantee of Variance-reduced Value Iteration

Theorem 5 (Sample Complexity of Sublinear Randomized Value Iteration [Sidford et al., 2018]).

In invocation of Sublinear Randomized Value Iteration (see Algorithm 3) requires

$$\tilde{\mathcal{O}}\left(|\mathcal{S}| \times |\mathcal{A}| \left(\frac{1}{\epsilon^2(1-\gamma)^4} + \frac{1}{(1-\gamma)^3}\right) \log\left(\frac{1}{\delta}\right)\right)$$

samples to obtain an ϵ-optimal value function with probability at least $1-\delta$, where $\epsilon \in (0, \frac{1}{1-\gamma}]$.

Comment on Variance-reduced Value Iteration

- There are two terms governing the sample complexity and the dominate term is task-dependent.

$$\tilde{\mathcal{O}}\left(\underbrace{\frac{1}{\epsilon^2(1-\gamma)^4}}_{\to\widehat{\mathcal{T}}}+\underbrace{\frac{1}{(1-\gamma)^3}}_{\to\widetilde{\mathcal{T}}}\right).$$

- To obtain an ϵ-optimal policy, the sample complexity becomes [Bertsekas and Tsitsiklis, 1996]

$$\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2(1-\gamma)^6}+\frac{1}{(1-\gamma)^3}\right)\Longrightarrow\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2(1-\gamma)^6}\right).$$

  - Only improve $1/(1-\gamma)$ over Phased Value Iteration (see Algorithm 2).

Proof Idea of Theorem 5

- `ApxVal` is almost same with the direct sample-average-approximation expect that the offset term is ϵ-optimal.
- With the same reasoning, the quality of `ApxVal` is reserved.

Lemma 6 (Quality of Approximate Value Operator).

With probability at least $1 - \delta$, the output of Approximate Value Operator (see Algorithm 5) satisfies that

$$||\bar{V} - \mathcal{T}(V)||_\infty \leq 2\gamma\epsilon.$$

Proof Idea of Theorem 5

- Based on Lemma 6, we can show that the quality of `SampledRandomizedVI` (i.e., the variance-reduced Bellman operator) is also maintained:

$$\begin{aligned} \|V_\ell - V^*\|_\infty &\leq \left\|V_\ell - V_\ell^\sharp\right\|_\infty + \left\|V_\ell^\sharp - V^*\right\|_\infty \\ &= ||V_\ell - \mathcal{T}(V_{\ell-1}) + \mathcal{T}(V_{\ell-1}) - V_\ell^\sharp|| + \left\|V_\ell^\sharp - V^*\right\|_\infty \\ &\leq 2\gamma\epsilon + \gamma \left\|V_{\ell-1} - V_{\ell-1}^\sharp\right\|_\infty + \gamma^\ell \|V_0 - V^*\|_\infty \\ &\leq \frac{2\gamma\epsilon}{1-\gamma} + \gamma^\ell \|V_0 - V^*\|_\infty , \end{aligned} \tag{7}$$

where $V_\ell^\sharp$ is the ℓ-th iterator of exact Value Iteration and we assume that $V_0 = V_0^\sharp$.

Proof Idea of Theorem 5

- By choosing a large enough iteration number L, e.g.,

$$L \geq \left\lceil \frac{1}{1-\gamma} \log \left(\frac{||V_0 - V^*||_\infty}{2\gamma\epsilon} \right) \right\rceil \quad \Longrightarrow \quad \gamma^\ell \left\| V_0 - V^* \right\|_\infty \leq \frac{2\gamma\epsilon}{1-\gamma}.$$

- Therefore, we have

$$\left\| V_\ell - V^* \right\|_\infty \leq \frac{2\gamma\epsilon}{1-\gamma} + \gamma^\ell \left\| V_0 - V^* \right\|_\infty \leq \frac{4\gamma\epsilon}{1-\gamma}. \tag{8}$$

- The sample complexity is computed in the next page.

Proof Idea of Theorem 5

- Sample complexity is consist of two terms: control-variate and the variance-reduced update.
- To obtain an ϵ-optimal control variate, we directly use Hoeffding's inequality:

$$n = \left\lceil \frac{2||V_0||_\infty^2}{\epsilon^2} \log\left(\frac{2L}{\delta}\right) \right\rceil \quad \Longrightarrow \quad n \sim \widetilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^2\epsilon^2}\right).$$

- To analyze the variance-reduced update, we need to upper bound the estimation range $\|V_\ell - V_0\|_\infty$.
- Importantly, the estimation range is reduced compared with the ordinary one $\|V_\ell\|_\infty$ in Phased Value Iteration.

Proof Idea of Theorem 5

- The estimation range $\|V_\ell - V_0\|_\infty$ is reduced over epochs.

$$
\begin{aligned}
\|V_\ell - V_0\|_\infty^2 &\leq \|V_\ell - V^* + V^* - V_0\|_\infty^2 \\
&\leq (||V_\ell - V^*||_\infty + ||V^* - V_0||_\infty)^2 \\
&\overset{(i)}{\leq} 2||V_\ell - V^*||_\infty^2 + 2||V^* - V_0||_\infty^2 \\
&\overset{(ii)}{\leq} 2\left(\frac{2\gamma\epsilon}{1-\gamma} + ||V_0 - V^*||_\infty\right)^2 + 2||V^* - V_0||_\infty^2 \\
&\leq \frac{8\epsilon^2}{(1-\gamma)^2} + 8||V_0 - V^*||_\infty^2,
\end{aligned}
\tag{9}
$$

- To conclude, the sample complexity of `SublinearRandomizedVI` is bounded by

$$
\mathcal{O}\left(|\mathcal{S}| \times |\mathcal{A}| \left[\frac{1}{(1-\gamma)^2\epsilon^2} + L\left(\frac{||V_0 - V^*||_\infty^2}{\epsilon^2} + \frac{1}{(1-\gamma)^2}\right)\right] \log\left(\frac{|\mathcal{S}| \times |\mathcal{A}| \times L}{\delta}\right)\right)
\tag{10}
$$

Proof Idea of Theorem 5

- By choosing $\epsilon_k = 0.5\epsilon_{k-1}$ in `SublinearRandomizedVI`, we infer that $||V_k - V^*||_\infty \leq \epsilon_k$ holds over epochs.

$$\mathcal{O}\left(|\mathcal{S}| \times |\mathcal{A}| \left[\frac{1}{(1-\gamma)^2\epsilon^2} + L\left(\cancel{\frac{||V_0 - V^*||_\infty^2}{\epsilon^2}} + \frac{1}{(1-\gamma)^2}\right)\right] \log\left(\frac{|\mathcal{S}| \times |\mathcal{A}| \times L}{\delta}\right)\right)$$

- By substituting $\epsilon := (1-\gamma)\epsilon$ in Equation (10) (due to the accuracy in Equation (8)), we note that the number of epochs K is a $1/(1-\gamma)$ independent term.
- Thus, the total sample complexity becomes:

$$\begin{aligned} &\implies \quad \tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^4\epsilon^2} + \frac{L}{(1-\gamma)^2}\right) \\ &\implies \quad \tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^4\epsilon^2} + \frac{1}{(1-\gamma)^3}\right). \end{aligned}$$

Variance-reduced Value Iteration (Refined)

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Q-Learning

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Speedy-Q-Learning

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Variance-reduced Q-Learning

Outline

- Background & Literature Review
- Model-free Methods
  - Phased Value Iteration
  - Variance-reduced Value Iteration
  - Q-Learning
  - Speedy-Q-Learning
  - Variance-reduced Q-Learning
- Model-based Methods
  - Model-based Value Iteration
- Summary

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Model-based Value Iteration

- On the other hand, the learner can first construct a virtual MDP $\widehat{\mathcal{M}}$ with collected samples, and then performs (population) Bellman operator on this recovered MDP $\widehat{\mathcal{M}}$.
- In this way, the learner does not need the iterative learning in Phased Value Iteration (see Algorithm 2), which requires new samples each iteration.

Model-based Value Iteration

Algorithm 6 Model-based Value Iteration

```
Input: $n$.
Collect $n$ next states for each state-action pair by calling the generative model $\mathcal{M}$.
Construct a virtual MDP with $\hat{P}$:
    $\hat{P}(s'|s,a) = \frac{\#\text{ times } (s,a) \mapsto s'}{n}.$
$\hat{V}^* \leftarrow$ Run Value Iteration (see Algorithm 1) on the virtual MDP.
Output: $\hat{V}^*$.
```

Theoretical Guarantee for Model-based Value Iteration

Theorem 7 (Sample Complexity of Model-based Value Iteration (Corase Analysis)).

Given a generative model $\mathcal{M}$, with the parameter:

$$n = \left\lceil \frac{1}{\epsilon^2 (1-\gamma)^4} \log\left(\frac{2 \times |\mathcal{S}| \times |\mathcal{A}|}{\delta} \right) \right\rceil,$$

Model-based Value Iteration (see Algorithm 6) can find a sub-optimal value function V_T such that $||V_T - V^*||_\infty \leq \epsilon$ from any initial solution V_0, where $\epsilon \in (0, \frac{1}{1-\gamma}]$ is the error tolerance with probability at least $1-\delta$, and the number of total samples used is

$$\mathcal{O}\left(\frac{|\mathcal{S}| \times |\mathcal{A}|}{\epsilon^2 (1-\gamma)^4} \log\left(\frac{|\mathcal{S}| \times |\mathcal{A}|}{\delta} \right) \right).$$

Proof Idea of Theorem 7

- Error decomposition:

$$\begin{aligned}\left\|V^* - \hat{V}^*\right\|_\infty &= \left\|\mathcal{T}(V^*) - \widehat{\mathcal{T}}(\hat{V}^*)\right\|_\infty \\ &= \left\|\mathcal{T}(V^*) - \mathcal{T}(\hat{V}^*) + \mathcal{T}(\hat{V}^*) - \widehat{\mathcal{T}}(\hat{V}^*)\right\|_\infty \\ &\le \gamma \left\|V^* - \hat{V}^*\right\|_\infty + \epsilon_n,\end{aligned}$$

  where we assume that the estimation error $\left\|\mathcal{T}(\hat{V}^*) - \widehat{\mathcal{T}}(\hat{V}^*)\right\|_\infty \le \epsilon_n$.

- Rearranging yields the bound:

$$\left\|V^* - \hat{V}^*\right\|_\infty \le \frac{\gamma}{1-\gamma}\epsilon_n. \tag{11}$$

Proof Idea of Theorem 7

- `Model-based Value Iteration` only requires the sample-average-approximation is accurate for $\hat{V}^*$.
- By substituting $\epsilon_n := \epsilon/(1-\gamma)$ in Equation (11), applying Hoeffding's inequality, we have

$$n \sim \mathcal{O}\left(\frac{1}{(1-\gamma)^4\epsilon^2}\right) \quad \Longrightarrow \quad \left\|V^* - \hat{V}^*\right\|_\infty \leq \epsilon.$$

Comment on Model-based Value Iteration

- `Model-based Value Iteration` only requires $\widehat{\mathcal{T}}$ to be accurate for $\hat{V}^*$.
  - On the contrast, `Phased Value Iteration` requires $\widehat{\mathcal{T}}$ to be accurate for each iterator $\hat{V}_t$.
- In this way, `Model-based Value Iteration` improves $\mathcal{O}(\frac{1}{1-\gamma})$ complexity compared to `Phased Value Iteration`.
- The above analysis for `Model-based Value Iteration` is coarse.
  - We independently bound the estimation error for each (s, a) pair then use a union bound.
  - For a single value function, the sequential structure of Bellman equation could provide a tighter bound to estimate the variance.

Refined Sample Complexity of Model-based Value Iteration

Theorem 8 (Refined Sample Complexity of Model-based Value Iteration [Azar et al., 2013]).

Given a generative oracle $\mathcal{M}$, with the parameter:

$$n = \mathcal{O}\left(\frac{1}{\epsilon^2(1-\gamma)^3}\log\left(\frac{2 \times |\mathcal{S}| \times |\mathcal{A}|}{\delta}\right)\right),$$

Model-based Value Iteration (see Algorithm 6) can find a sub-optimal value function V_T such that $||V_T - V^*||_\infty \leq \epsilon$ from any initial solution V_0, where $\epsilon \in (0, \frac{1}{1-\gamma}]$ is the error tolerance with probability at least $1-\delta$, and the number of total samples used is

$$\mathcal{O}\left(\frac{|\mathcal{S}| \times |\mathcal{A}|}{\epsilon^2(1-\gamma)^3}\log\left(\frac{|\mathcal{S}| \times |\mathcal{A}|}{\delta}\right)\right).$$

Proof Idea of Theorem 8

- We start with the general form of the error decomposition.
- Note that $\hat{V}^* \geq \hat{V}^{\pi^*}$, we have

$$\begin{aligned}
V^* - \hat{V}^* &\leq V^* - \hat{V}^{\pi^*} \\
&= \left(\mathbb{I} - \gamma P^{\pi^*}\right)^{-1} r^{\pi^*} - \left(\mathbb{I} - \gamma \hat{P}^{\pi^*}\right)^{-1} r^{\pi^*} \qquad \left(r^{\pi^*}(s) = r\left(s, \pi^*(s)\right)\right) \\
&= \left(\mathbb{I} - \gamma \hat{P}^{\pi^*}\right)^{-1} \left[\left(\mathbb{I} - \gamma \hat{P}^{\pi^*}\right) - \left(\mathbb{I} - \gamma P^{\pi^*}\right)\right] \left(\mathbb{I} - \gamma P^{\pi^*}\right)^{-1} r^{\pi^*} \\
&= \gamma \left(\mathbb{I} - \gamma \hat{P}^{\pi^*}\right)^{-1} \left[P^{\pi^*} - \hat{P}^{\pi^*}\right] \left(\mathbb{I} - \gamma P^{\pi^*}\right)^{-1} r^{\pi^*} \\
&= \gamma \underbrace{\left(\mathbb{I} - \gamma \hat{P}^{\pi^*}\right)^{-1}}_{\text{precondition}} \underbrace{\left[P^{\pi^*} - \hat{P}^{\pi^*}\right] V^*,}_{\text{estimation}}
\end{aligned}$$

where $\leq$ holds elementwise.

Proof Idea of Theorem 8

- Similarly, we have the elementwise-error bound:

$$\begin{cases} V^* - \hat{V}^* \leq \gamma \left(\mathbb{I} - \gamma \hat{P}^{\pi^*} \right)^{-1} \left[P^{\pi^*} - \hat{P}^{\pi^*} \right] V^* \\ V^* - \hat{V}^* \geq \gamma \left(\mathbb{I} - \gamma \hat{P}^{\hat{\pi}^*} \right)^{-1} \left[P^{\pi^*} - \hat{P}^{\pi^*} \right] V^*. \end{cases}$$

- To apply Bernstein's inequality, we need to consider the variance of estimation.

$$\sigma^2_{V^\pi}(s,a) = \gamma^2 \mathbb{V}_{s' \sim p(\cdot|s,a)} \left[V^\pi \left(s' \right) \right], \quad \text{and} \quad \hat{\sigma}^2_{V^*}(s,a) = \gamma^2 \mathbb{V}_{s' \sim \hat{p}(\cdot|s,a)} \left[V^\pi \left(s' \right) \right].$$

- Furthermore, we need to unify the RHS to get a single variance bound (see the next page).

Proof Idea of Theorem 8

Lemma 9 (Elementwise bounds on σ_{V^*}).

With probability at least $1-\delta$, the following relations hold separately:

$$\sigma_{V^*} \leq \hat{\sigma}_{\hat{V}^{\pi^*}} + b_v \vec{1},$$
$$\sigma_{V^*} \leq \hat{\sigma}_{\hat{V}^{\hat{\pi}^*}} + b_v \vec{1},$$

where b_v is defined as

$$b_v = \left(\frac{18\gamma^4 \log\left(\frac{3|\mathcal{S}|\times|\mathcal{A}|}{\delta}\right)}{n(1-\gamma)^4} \right)^{1/4} + \sqrt{\frac{4\gamma^2 \log\left(\frac{6|S|\times|\mathcal{A}|}{\delta}\right)}{n(1-\gamma)^4}}$$

Proof Idea of Theorem 8

- With Lemma 9, we can bound the estimation error.

Lemma 10 (Elementwise bounds on $(P^{\pi^*} - \hat{P}^{\pi^*})V^*$).

Define the following constants:

$$\begin{aligned} c_{pv} &= 2\log\left(\tfrac{2|\mathcal{S}|\times|\mathcal{A}|}{\delta}\right) \\ b_{pv} &= \left(\frac{5\log\left(\frac{6|\mathcal{S}|\times|\mathcal{A}|}{\delta}\right)}{n}\left(\frac{\gamma}{1-\gamma}\right)^{4/3}\right)^{3/4} + \frac{4\log\left(\frac{12|S|\times|\mathcal{A}|}{\delta}\right)}{n(1-\gamma)^2}. \end{aligned}$$

With probability at least $1-\delta$, we have

$$-\sqrt{\frac{c_{pv}\hat{\sigma}^2_{\hat{V}^{\pi^*}}}{n}} - b_{pv}\vec{1} \leq \gamma\left(P^{\pi^*} - \hat{P}^{\pi^*}\right)V^* \leq \sqrt{\frac{c_{pv}\hat{\sigma}^2_{\hat{V}^{\pi^*}}}{n}} + b_{pv}\vec{1}$$

Proof Idea of Theorem 8

- Finally, we need to consider the multiplication by the precondition matrix and the variance.

$$\left(\mathbb{I} - \gamma \hat{P}^{\pi^*}\right)^{-1} \left[P^{\pi^*} - \hat{P}^{\pi^*}\right] V^* \approx \left(\mathbb{I} - \gamma \hat{P}^{\pi^*}\right)^{-1} \hat{\sigma}_{\hat{V}^{\pi^*}}.$$

- A naive bound with Cauchy-Schwarz inequality inequality is $\mathcal{O}\left(\frac{1}{(1-\gamma)^2}\right)$.
  - Together with $\sqrt{\frac{1}{n}}$ in Lemma 10, this yields the sample complexity of $\mathcal{O}\left(\frac{1}{(1-\gamma)^4}\right)$.
- However, we will show that the sequential structure of MDP yields a tighter bound of $\mathcal{O}\left(\frac{1}{(1-\gamma)^{1.5}}\right)$.

Proof Idea of Theorem 8

- Consider the "total variance":

$$\Sigma^{\pi}(s,a) = \mathbb{E}\left[\left\{\sum_{t\geq 0}\gamma^{t} r\left(s_t,a_t\right) - Q^{\pi}(s,a)\right\}^{2} \mid s_0 = s, a_0 = a\right].$$

- We show that "total variance" satisfies the following Bellman equation.

Lemma 11 (Bellman-like variance).

Σ^{π} satisfies the following Bellman equation:

$$\Sigma^{\pi} = \sigma^{2}_{V^{\pi}} + \gamma^{2} P^{\pi} \Sigma^{\pi}.$$

Proof Idea of Theorem 8

- With Lemma 11, we get a tighter bound:

$$\left\|\left(\mathbb{I}-\gamma^2 P^\pi\right)^{-1}\sigma_{V^\pi}^2\right\|_\infty = \|\Sigma^\pi\|_\infty \leq \frac{1}{(1-\gamma)^2}$$

$$\left\|\left(\mathbb{I}-\gamma P^\pi\right)^{-1}\sigma_{V^\pi}\right\|_\infty \leq 2\|\sqrt{\frac{1}{1-\gamma}\Sigma^\pi}\|_\infty \leq \frac{2}{(1-\gamma)^{1.5}}.$$

- In this way, we prove that the total sample complexity is $\mathcal{O}\left(\frac{1}{(1-\gamma)^3\epsilon^2}\right)$.

Outline

Background & Literature Review

Model-free Methods

- Phased Value Iteration
- Variance-reduced Value Iteration
- Q-Learning
- Speedy-Q-Learning
- Variance-reduced Q-Learning

Model-based Methods

- Model-based Value Iteration

Summary

Summary

- `Model-free` methods have the following properties:
  - they iteratively collect samples and update the value function.
  - variance-reduction plays an important role.
- `Model-based` methods have the following properties:
  - they collect enough samples once and solve the optimal value function.
  - it's easy to achieve the minimax-optimal bound for both policy/(value function).
- It's non-trivial to obtain an ϵ-optimal policy from an imperfect value function.

Prior Art

Sample Complexity to obtain an ϵ-optimal policy

algorithm	sample size range	sample complexity	ε-range
empirical QVI Azar et al. '13	$\left[\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert}{(1-\gamma)^2}, \infty\right)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3\varepsilon^2}$	$(0, \frac{1}{\sqrt{(1-\gamma)\lvert\mathcal{S}\rvert}}]$
sublinear randomized VI Sidford et al. '18a	$\left[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^2}, \infty\right)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^4\varepsilon^2}$	$\left(0, \frac{1}{1-\gamma}\right]$
variance-reduced QVI Sidford et al. '18b	$\left[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3}, \infty\right)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3\varepsilon^2}$	$(0, 1]$
randomized primal-dual Wang '17	$\left[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^2}, \infty\right)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^4\varepsilon^2}$	$(0, \frac{1}{1-\gamma}]$
empirical MDP + planning Agarwal et al. '19	$\left[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^2}, \infty\right)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3\varepsilon^2}$	$(0, \frac{1}{\sqrt{1-\gamma}}]$

Table from `http://www.stat.cmu.edu/~ytwei/documents/slides/model-based-rl-slides.pdf`.

Open Problem

- Q1: breaking the sample barrier for model-free methods with a generative oracle.
- Q2: online model-free learning with variance-reduction.

References I

M. G. Azar, R. Munos, and H. J. Kappen. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. Machine Learning, 91(3):325–349, 2013.

D. P. Bertsekas and J. Tsitsiklis. Neuro-Dynamic Programming. Athena Scientific, 1996.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems, pages 315–323, 2013.

M. J. Kearns and S. P. Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In Advances in Neural Information Processing Systems, pages 996–1002, 1999.

M. L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Statistics. Wiley, 1994.

References II

A. Sidford, M. Wang, X. Wu, and Y. Ye. Variance reduced value iteration and faster algorithms for solving markov decision processes. In Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 770–787, 2018.

R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. MIT press, 2018.