Final Report: Near-Optimal Time and Sample Complexities for Solving Discounted Markov Decision Process with a Generative Model

Andrea Baretta, Christopher Seybold

June 14, 2024

1 Introduction

A classic problem in reinforcement learning is to find an optimal policy given a full Markov Decision Process as an input. In this case, the probabilistic transition function is fully known, and algorithms exist with guaranteed convergence to an exact solution.

A fundamental problem with this approach, however, is that in many real-world scenarios, the agent does not have complete knowledge of the environment and must instead learn from experience. Thus, the motivation behind our paper of interest is to avoid making this assumption, and instead compute an ϵ-optimal policy of a discounted Markov Decision Process with high probability and with as low a sample and time complexity as possible, where a generative sampling model is required to access its transition function. The authors then extend their methods to finite-horizon MDPs and give a nearly matching sample complexity lower bound. It is also important to note that the authors assume that the agent can sample any state-action pair from the transition function in O(1) time.

1.1 Generative Models

Generative models have emerged as a powerful tool in the field of reinforcement learning, providing a mechanism to approximate the dynamics of the environment, thereby greatly reducing the computational costs for predicting outcomes of sequences of actions. These models aim to learn the true data distribution of a training set in order to generate augmented data, which provides a fast, cheap, and safe estimate that can be used by an RL algorithm to limit the interaction required with the actual environment.

This is particularly beneficial in scenarios where interactions with the environment are costly or potentially harmful. For example, it is completely unfeasible to send an untrained autonomous vehicle out onto the streets in hopes that it will learn from its mistakes, which brings about the need for techniques to approximate a real environment.

1.2 Model-Based vs Model-Free RL

Model-Based RL: In model-based RL, the agent maintains an explicit model of the underlying MDP. This model includes information about the transition dynamics (how the environment evolves from one state to another) and the reward function (the immediate reward associated with each state-action pair). The agent uses this model to simulate possible trajectories and evaluate different policies, allowing it to make informed decisions as to maximize future reward. Model-based methods are often more sample-efficient because they can directly access and utilize the structured model to explore potential outcomes before taking actions. However, creating an accurate model can be challenging, especially when dealing with complex or partially observable environments.

Model-Free RL: In contrast, model-free algorithms do not explicitly maintain a model of the MDP, and instead learn directly from interaction with the environment by estimating a value function or policy. Examples of model-free algorithms include Q-learning, where the agent uses a Q function to estimate the expected sum of future rewards to select optimal actions; policy gradient methods, where the agent directly optimizes the parameters of its policy by running gradient descent with a gradient estimated based on previously received rewards; and the sublinear randomized QVI algorithm proposed in the examined publication.

2 Applications and Limitations

In addition to highlighting the significant contributions that the authors made, it is crucial to acknowledge the limitations of the methods presented, as certain assumptions and conditions limit their applicability and efficiency in complex, real-world scenarios. These constraints, however, offer tremendous opportunities for future research and refinement of the proposed methods.

2.1 Transition Dynamics

Transition dynamics are essentially the rules that govern how the environment changes in response to an agent's actions. They can be deterministic, where the next state is fully determined by the current state and action, or stochastic, where the next state is probabilistic and depends on the current state and action. Learning the transition dynamics is a key part of RL algorithms, particularly those that are model-based, as they help the agent predict the outcomes of its actions and plan accordingly. However, as the complexity of the agents environment increases, these dynamics become much harder to efficiently and accurately capture.

2.2 Computational Complexity

The algorithm presented in the paper by [SWW+18] has a sample and time complexity that is proportional to the number of states and actions. It follows that if the number of states and actions significantly increases, the time required for the algorithm to run does the same. In complex environments that robots and autonomous vehicles must be capable of navigating, this could lead to computational challenges.

The assumption that the generative model can sample from the transition function in O(1) time is a key aspect of the algorithm presented in our paper of interest, so it is important to note that the size of the state and action spaces directly impacts the feasibility of this assumption. Attempting to sample from the transition function in a large space associated with a highly complex environment could potentially take much more than constant time, especially if the transition dynamics are complex and difficult to model. For example, the next state of a self-driving car could depend on its current speed, direction, the positions of other cars, traffic signals, road conditions, and so on.

2.3 Complex vs Simple Problems

In a self-driving car scenario, the state and action spaces are incredibly large. For instance, the state could include the positions of all nearby cars, pedestrians, traffic lights, road signs, and even weather conditions. The action space could include all possible combinations of steering angles, acceleration, and braking.

On the other hand, a game like checkers has a much smaller state and action space. The states can be represented by the positions of the pieces on the board, and the actions are the legal moves for each piece. Therefore, the algorithm can handle the complexity of a game like checkers more feasibly.

2.4 Resolution

While the algorithm presented in the paper is theoretically sound and can be applied to certain problems, its practical application to complex tasks like self-driving cars is limited due to the large state and action spaces. However, it could be quite effective for simpler tasks with smaller state and action spaces, like a game playing agent for checkers.

3 Previous State-of-the-Art

Our paper of interest cleverly builds on the results of a handful of other works that were at the forefront of reinforcement learning research at their time of publication. By piecing together these previous results with their own unique insights, the authors of our examined paper were able to prove an optimal theoretical lower bound for computing an ϵ-optimal policy of a discounted MDP where the transition function is accessed through a generative sampling model.

3.1 Minimax PAC Bounds on the Sample Complexity of Reinforcement Learning with a Generative Model

The focal point of [AMK13] that our discussed paper builds on involves model-based RL algorithms for finite state-action problems, assuming access to a generative model of the MDP. The authors provide a sample complexity lower bound on the order of $\tilde{O}((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$ for finding an ϵ-optimal policy using an algorithm called empirical Q-value iteration. However, to achieve this lower bound, the authors had to make assumptions about the size of ϵ, specifically that $\epsilon = \tilde{O}([(1-\gamma)|\mathcal{S}|]^{-1/2})$. So, while it served as a nice baseline for a theoretical lower bound on sample complexity, it made no efforts toward bounding time complexity, and future research was required to remove any assumptions on the size of epsilon.

3.2 Variance Reduced Value Iteration and Faster Algorithms for Solving Markov Decision Processes

The notable progress that [SWWY18] made was improving the previous optimal bounds on the time complexity by a factor of $(1-\gamma)^{-1}$ in regards to computing an ϵ-optimal policy using a generative sampling model. With an algorithm called Sublinear Randomized Value Iteration, the authors achieved a lower bound on the order of $\tilde{O}((1-\gamma)^{-4}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$, while making no assumptions about the size of epsilon. However, they did not provide any bound on the sample complexity, and were still a factor of $(1-\gamma)^{-1}$ behind what was thought possible based on current literature, which our paper in question would achieve shortly there after.

4 Technical Result

4.1 Algorithm

The main algorithm presented by the researchers takes as input a policy and value function pair $(\pi^{(0)}, \boldsymbol{v}^{(0)})$, an upper bound $u \in [0, (1-\gamma)^{-1}]$ on the error of the value function, and an error probability $\delta \in (0,1)$. The algorithm further requires the value function to meet the monotonicity condition, i.e. that $\boldsymbol{v}^{(0)} \leq \mathcal{T}_{\pi^{(0)}}(\boldsymbol{v}^{(0)})$. The algorithm then outputs a policy and value function pair $(\pi, \boldsymbol{v})$ that satisfies the montonicity condition and that is at least $\frac{u}{2}$-optimal with probability at least $1-\delta$. A meta-algorithm then iteratively calls the main algorithm until error is reduced to under a desired amount with high probabibility.

The error-halving algorithm makes use of three key techniques. The first two of these are called the *monotonicity technique* and *variance reduction*. The article augments these with a novel third technique, the *total variance technique*. The algorithm presented by the authors is a non-trivial integration of these three techniques.

The Monotonicity Technique This technique leverages the monotonicity of the Bellman operator. In classic value iteration, we repeatedly apply the Bellman operator on the value function until we are ϵ-close to the optimal policy.

$$\boldsymbol{v}^{(i)}(s) \leftarrow \max_a (r(s,a) + \gamma \boldsymbol{P}_{s,a}^{\top} \boldsymbol{v}^{(i-1)}) \tag{1}$$

However, the value function of the corresponding greedy policy is only $\epsilon(1-\gamma)^{-1}$ optimal. To avoid the $(1-\gamma)^{-1}$ increase in error, the value iteration algorithm is augmented by enforcing monotonicity.

Algorithm 1 Variance-Reduced QVI

```
1: Input: A sampling oracle for DMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \boldsymbol{r}, \boldsymbol{P}, \gamma)$
2: Input: Upper bound on error $u \in [0, (1-\gamma)^{-1}]$ and error probability $\delta \in (0, 1)$
3: Input: Initial values $\boldsymbol{v}^{(0)}$ and policy $\pi^{(0)}$ such that $\boldsymbol{v}^{(0)} \le \mathcal{T}_{\pi^{(0)}}\boldsymbol{v}^{(0)}$, and $\boldsymbol{v}^* - \boldsymbol{v}^{(0)} \le u\mathbf{1}$;
4: Output: $\boldsymbol{v}, \pi$ such that $\boldsymbol{v} \le \mathcal{T}_\pi(\boldsymbol{v})$ and $\boldsymbol{v}^* - \boldsymbol{v} \le (u/2) \cdot \mathbf{1}$.
5:
6: INITIALIZATION:
7: Let $\beta \leftarrow (1-\gamma)^{-1}$, and $R \leftarrow \lceil c_1 \beta \ln[\beta u^{-1}] \rceil$ for constant $c_1$;
8: Let $m_1 \leftarrow c_2 \beta^3 u^{-2} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$ for constant $c_2$;
9: Let $m_2 \leftarrow c_3 \beta^2 \log[2R|\mathcal{S}||\mathcal{A}|\delta^{-1}]$ for constant $c_3$;
10: Let $\alpha_1 \leftarrow {m_1}^{-1} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$;
11: For each $(s,a) \in \mathcal{S} \times \mathcal{A}$, sample independent samples $s_{s,a}^{(1)}, s_{s,a}^{(2)}, \ldots, s_{s,a}^{(m_1)}$ from $\boldsymbol{P}_{s,a}$;
12: Initialize $\boldsymbol{w} = \tilde{\boldsymbol{w}} = \widehat{\boldsymbol{\sigma}} = \boldsymbol{Q}^{(0)} \leftarrow \mathbf{0}_{\mathcal{S}\times\mathcal{A}}$, and $i \leftarrow 0$;
13: for each $(s,a) \in \mathcal{S} \times \mathcal{A}$ do
14:     \\Compute empirical estimates of $\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}$ and $\boldsymbol{\sigma}_{\boldsymbol{v}^{(0)}}(s,a)$
15:     Let $\tilde{\boldsymbol{w}}(s,a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} \boldsymbol{v}^{(0)}(s_{s,a}^{(j)})$
16:     Let $\widehat{\boldsymbol{\sigma}}(s,a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} (\boldsymbol{v}^{(0)})^2(s_{s,a}^{(j)}) - \tilde{\boldsymbol{w}}^2(s,a)$
17:
18:     \\Shift the empirical estimate to have one-sided error and guarantee monotonicity
19:     $\boldsymbol{w}(s,a) \leftarrow \tilde{\boldsymbol{w}}(s,a) - \sqrt{2\alpha_1 \widehat{\boldsymbol{\sigma}}(s,a)} - 4\alpha_1^{3/4} \|\boldsymbol{v}^{(0)}\|_\infty - (2/3)\alpha_1 \|\boldsymbol{v}^{(0)}\|_\infty$
20:
21:     \\Compute coarse estimate of the Q-function
22:     $\boldsymbol{Q}^{(0)}(s,a) \leftarrow \boldsymbol{r}(s,a) + \gamma \boldsymbol{w}(s,a)$
23: end for
24:
25: REPEAT:            \\successively improve
26: for $i = 1$ to $R$ do
27:     \\Compute $\boldsymbol{g}^{(i)}$ the estimate of $\boldsymbol{P}\big[\boldsymbol{v}^{(i)} - \boldsymbol{v}^{(0)}\big]$ with one-sided error
28:     Let $\boldsymbol{v}^{(i)} \leftarrow \boldsymbol{v}(\boldsymbol{Q}^{(i-1)})$, $\pi^{(i)} \leftarrow \pi(\boldsymbol{Q}^{(i-1)})$; \\let $\tilde{\boldsymbol{v}}^{(i)} \leftarrow \boldsymbol{v}^{(i)}, \tilde{\pi}^{(i)} \leftarrow \pi^{(i)}$ (for analysis);
29:     For each $s \in \mathcal{S}$, if $\boldsymbol{v}^{(i)}(s) \le \boldsymbol{v}^{(i-1)}(s)$, then $\boldsymbol{v}^{(i)}(s) \leftarrow \boldsymbol{v}^{(i-1)}(s)$ and $\pi^{(i)}(s) \leftarrow \pi^{(i-1)}(s)$;
30:     For each $(s,a) \in \mathcal{S} \times \mathcal{A}$, draw independent samples $\tilde{s}_{s,a}^{(1)}, \tilde{s}_{s,a}^{(2)}, \ldots, \tilde{s}_{s,a}^{(m_2)}$ from $\boldsymbol{P}_{s,a}$;
31:     Let $\boldsymbol{g}^{(i)}(s,a) \leftarrow \frac{1}{m_2} \sum_{j=1}^{m_2} \big[\boldsymbol{v}^{(i)}(\tilde{s}_{s,a}^{(j)}) - \boldsymbol{v}^{(0)}(\tilde{s}_{s,a}^{(j)})\big] - (1-\gamma)u/8$;
32:
33:     \\Improve $\boldsymbol{Q}^{(i)}$
34:     $\boldsymbol{Q}^{(i)} \leftarrow \boldsymbol{r} + \gamma \cdot [\boldsymbol{w} + \boldsymbol{g}^{(i)}]$;
35: end for
36: return $\boldsymbol{v}^{(R)}, \pi^{(R)}$.
```

This is done by computing the greedy policy $\pi^{(i)}$ at every iteration.

$$\pi^{(i+1)}(s) \leftarrow \arg\max_a (r(s,a) + \gamma \boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(i)}) \tag{2}$$

Then, we enforce that, element-wise, with high probability,

$$\boldsymbol{v}^{(i+1)} \le \mathcal{T}_{\pi^{(i+1)}}(\boldsymbol{v}^{(i+1)}) \tag{3}$$

If this condition is enforced, then

$$\boldsymbol{v}^{(N)} \le \mathcal{T}_{\pi^{(N)}}(\boldsymbol{v}^{(N)}) \le \mathcal{T}_{\pi^{(N)}}^2(\boldsymbol{v}^{(N)}) \le \ldots \tag{4}$$

$$\le \mathcal{T}_{\pi^{(N)}}^\infty(\boldsymbol{v}^{(N)}) = \boldsymbol{v}^{\pi^{(N)}} \le \boldsymbol{v}^* \tag{5}$$

This implies that after any number N of iterations, the value function of the greedy policy, $\boldsymbol{v}^{\pi^{(N)}}$, will be at least as good as $\boldsymbol{v}^{(N)}$.

The authors of [SWW+18] base their approach on that of [SWWY18], which proposes that if one value iteration improves a value function $\boldsymbol{v}$ by more than $2\gamma\epsilon$, then that quantity should be subtracted from the value function.

$$\boldsymbol{v}^{(i+1)} \leftarrow \boldsymbol{v}^{(i+1)} - \mathbb{1}_{\{\boldsymbol{v}^{(i+1)} - 2\gamma\epsilon > \boldsymbol{v}^{(i)}\}} \cdot 2\gamma\epsilon \tag{6}$$

However, [SWW+18] enforces monotonicity in more than ways than one. On Line 29 of the algorithm, if a value iteration decreases the value function $\boldsymbol{v}^{(i)}$, then the iteration is simply discarded, and $\boldsymbol{v}^{(i-1)}$ is used. Further, similarly to [SWWY18], at each value iteration, $(1-\gamma)\epsilon/8$ is subtracted from the estimated $\boldsymbol{Q}$-function on Line 31. Line 19 of the initialization phase of the algorithm also contributes towards ensuring monotonicity, though this will be explained in greater detail in the next section.

The Variance Reduction Technique In classical empirical value iteration, we build an empirical estimator of $\boldsymbol{P}_{s,a}^{\top}\boldsymbol{v}^{(i)}$ for every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. The goal of variance reduction is to estimate a lower variance variable than $\boldsymbol{P}_{s,a}^{\top}\boldsymbol{v}^{(i)}$ so as to decrease the number of samples needed to achieve a good error bound. Suppose we begin with $\boldsymbol{v}^{(0)}$ being an ϵ-optimal value function. By the Hoeffding bound, to achieve a $\frac{\epsilon}{2}$-optimal policy, it takes $\tilde{O}((1-\gamma)^{-4}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$ samples per iteration to estimate $\boldsymbol{P}_{s,a}^{\top}\boldsymbol{v}^{(i)}$ and a further factor of $\tilde{O}((1-\gamma)^{-1})$ for value iteration to converge ([SWWY18]), resulting in a $\tilde{O}((1-\gamma)^{-5}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$ sample complexity. To reduce this, we can begin by obtaining an initial, accurate estimate of $\boldsymbol{P}_{s,a}^{\top}\boldsymbol{v}^{(0)}$. Then, we can estimate $\boldsymbol{P}_{s,a}^{\top}(\boldsymbol{v}^{(i)} - \boldsymbol{v}^{(0)})$ with $\tilde{O}((1-\gamma)^{-2})$ samples. Lastly, we update the value function with the following rule:

$$\boldsymbol{v}^{(i+1)}(s) \leftarrow \max_a \left(r(s,a) + \gamma \left[\boldsymbol{P}_{s,a}^{\top}(\boldsymbol{v}^{(i)} - \boldsymbol{v}^{(0)}) + \boldsymbol{P}_{s,a}^{\top}\boldsymbol{v}^{(0)} \right] \right) \tag{7}$$

As before, it takes a further factor $\tilde{O}((1-\gamma)^{-1})$ samples per iteration to converge. The final sample complexity is $\tilde{O}((1-\gamma)^{-5}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$.

The reason variance reduction works so well is that for $z_{s,a} \sim \boldsymbol{P}_{s,a}$, in the worst case, the variance of $\boldsymbol{P}_{s,a}^{\top}\boldsymbol{v}^{(i)}$ is upper bounded by:

$$\begin{align} \mathrm{Var}(\boldsymbol{v}^{(i)}(z_{s,a})) &\leq \|\boldsymbol{v}^{(i)}\|_\infty^2 \tag{8} \\ &\leq \frac{1}{(1-\gamma)^2} \tag{9} \end{align}$$

On the other hand, thanks to monotonicity,

$$\begin{align} \mathrm{Var}((\boldsymbol{v}^{(i)} - \boldsymbol{v}^{(0)})(z_{s,a})) &\leq \|\boldsymbol{v}^{(i)} - \boldsymbol{v}^{(0)}\|_\infty^2 \tag{10} \\ &\leq \|\boldsymbol{v}^{*} - \boldsymbol{v}^{(0)}\|_\infty^2 \tag{11} \\ &\leq \epsilon_n^2 \tag{12} \\ &= \frac{\epsilon^2}{2^{2n}} \tag{13} \end{align}$$

where ϵ is the error of the first guess of the value function, and n is the number of times the error halving algorithm has already been called. It is easy to see that the variance of the variable we are trying to empirically estimate decreases exponentially with n. Consequently, the number of samples needed to confidently estimate $\boldsymbol{P}_{s,a}^{\top}(\boldsymbol{v}^{(i)} - \boldsymbol{v}^{(0)})$ also decreases. As the algorithm is called more, when the value functions are closer to optimal, variance reduction becomes more and more powerful.

In the algorithm, variance reduction is used in the value iteration phase of the algorithm. On Line 31, $\boldsymbol{g}^{(i)}$ is computed as the estimator of $\boldsymbol{P}_{s,a}^{\top}(\boldsymbol{v}^{(i)} - \boldsymbol{v}^{(0)})$, and on Line 34 we update the $\boldsymbol{Q}$-function according to Eqn. 7. In addition to this, to help ensure monotonicity, a pessimistic initial estimate of $\boldsymbol{P}_{s,a}^{\top}\boldsymbol{v}^{(0)}$ is computed on Line 19 by taking the unbiased estimator and subtracting a component proportional to its variance and a component proportional to its l-∞ norm.

The Total Variance Technique When combining the previous two methods, [SWWY18] sets the update error at each iteration to be at most $\frac{\epsilon}{2}(1-\gamma)$ in order to compensate for error accumulated over $\tilde{O}((1-\gamma)^{-1})$ iterations. However, the authors demonstrate that by using the Bernstein inequality, the update error can be relaxed, and the sample complexity decreased. Let

$$\boldsymbol{\sigma}_\pi(s) = \mathrm{Var}_{s' \sim \boldsymbol{P}_{s,\pi(s)}}(\boldsymbol{v}^\pi(s')) \tag{14}$$

Then it can be shown that the accumulated error is proportional to the following expression, up to log factors:

$$\text{acc. error} \propto \sum_{i=0}^{\infty} \gamma^i \boldsymbol{P}_\pi^i \sqrt{\boldsymbol{\sigma}_\pi / m} \leq c_1 \left(\frac{1}{1-\gamma} \sum_{i=0}^{\infty} \gamma^{2i} \boldsymbol{P}_\pi^i \boldsymbol{\sigma}_\pi / m \right)^{\frac{1}{2}} \tag{15}$$

The authors took the proportional relationship from an analysis by [AMK13], and they arrive at the inequality by using the Cauchy-Schwartz inequality. Here, m is the number of samples taken from the generative model for each value iteration, and c_1 is some constant. In order to bound this expression, first consider

$$\boldsymbol{\Sigma}^\pi(s,a) = \mathrm{Var}[\boldsymbol{Q}^\pi(s,a)] \tag{16}$$

From the law of total variance, it can be shown that $\boldsymbol{\Sigma}^\pi$ satisfies a Bellman equation of the form

$$\boldsymbol{\Sigma}^\pi = \gamma^2 \boldsymbol{\sigma}_{\boldsymbol{v}^\pi} + \gamma^2 \boldsymbol{P}_\pi \boldsymbol{\Sigma}^\pi \tag{17}$$

Infinitely expanding the Bellman equation, we get the following expression for $\boldsymbol{\Sigma}^\pi$:

$$\boldsymbol{\Sigma}^\pi = \gamma^2 \boldsymbol{\sigma}_{\boldsymbol{v}^\pi} + \gamma^2 \boldsymbol{P}_\pi \left(\gamma^2 \boldsymbol{\sigma}_{\boldsymbol{v}^\pi} + \gamma^2 \boldsymbol{P}_\pi \left(\gamma^2 \boldsymbol{\sigma}_{\boldsymbol{v}^\pi} + \gamma^2 \boldsymbol{P}_\pi (\ldots) \right) \right) \tag{18}$$

$$\boldsymbol{\Sigma}^\pi = \gamma^2 \sum_{i=0}^{\infty} \gamma^{2i} \boldsymbol{P}_\pi^i \boldsymbol{\sigma}_{\boldsymbol{v}^\pi} \tag{19}$$

Finally, by the definition of $\boldsymbol{\Sigma}^\pi$, we can bound it by $(1-\gamma)^{-2}$:

$$\boldsymbol{\Sigma}^\pi(s,a) = \mathrm{Var}[\boldsymbol{Q}^\pi(s,a)] = \mathbb{E}[\boldsymbol{Q}^\pi(s,a)^2] - \mathbb{E}[\boldsymbol{Q}^\pi(s,a)]^2 \tag{20}$$

$$\leq \mathbb{E}[\boldsymbol{Q}^\pi(s,a)^2] \leq \frac{1}{(1-\gamma)^2} \tag{21}$$

We can then bound the error accumulated from value iteration by

$$c_1 \left(\frac{1}{1-\gamma} \sum_{i=0}^{\infty} \gamma^{2i} \boldsymbol{P}_\pi^i \boldsymbol{\sigma}_\pi / m \right)^{\frac{1}{2}} = \tilde{O} \left(\frac{1}{m(1-\gamma)^3} \right)^{\frac{1}{2}} \tag{22}$$

Picking the number of samples $m = \tilde{O}((1-\gamma)^{-3}\epsilon^{-2}|\mathcal{S}||\mathcal{A}|)$, we can control error. Then, this brings the final sample complexity further down by another factor of $\tilde{O}((1-\gamma)^{-1})$.

In the algorithm, the total variance reduction analysis is used to inform the number of samples used to build the estimator of $\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}$. This can be seen on Lines 7 and 8.

4.2 Lemma 5.1

Lemma 5.1 helps establishes an error bound on the algorithm's final estimator of $\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}$. This allows us to find a lower bound with high probability, something necessary for both variance reduction and monotonicity. In this section, we present a derivation of the lemma.

Lemma 5.1 (Empirical Estimation Error). *Let $\tilde{\boldsymbol{w}}$ and $\widehat{\boldsymbol{\sigma}}$ be computed in Line 15 and 16 of Algorithm 1. Recall that $\tilde{\boldsymbol{w}}$ and $\widehat{\boldsymbol{\sigma}}$ are empirical estimates of $\boldsymbol{P}\boldsymbol{v}$ and $\boldsymbol{\sigma}_{\boldsymbol{v}} = \boldsymbol{P}\boldsymbol{v}^2 - (\boldsymbol{P}\boldsymbol{v})^2$ using m_1 samples per (s,a) pair. With probability at least $1-\delta$, for $L \overset{\text{def}}{=} \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})$ and $m_1 \geq 2L$, we have*

$$\left| \tilde{\boldsymbol{w}} - \boldsymbol{P}^\top \boldsymbol{v}^{(0)} \right| \leq \sqrt{2 m_1^{-1} \boldsymbol{\sigma}_{\boldsymbol{v}^{(0)}} \cdot L} + 2(3m_1)^{-1} \|\boldsymbol{v}^{(0)}\|_\infty L \tag{23}$$

and

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad \left|\widehat{\boldsymbol{\sigma}}(s,a) - \boldsymbol{\sigma}_{\boldsymbol{v}^{(0)}}(s,a)\right| \leq 4\|\boldsymbol{v}^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1}L}. \tag{24}$$

Before we begin with the derivation, it is useful to first recall a couple of important theorems. Let $\boldsymbol{v} \in \mathbb{R}^{\mathcal{S}}$ be a vector, and let $\boldsymbol{p} \in \Delta_{\mathcal{S}}$ be a probability vector. Let $\boldsymbol{p}_m \in \Delta_{\mathcal{S}}$ be the empirical estimator of $\boldsymbol{p}$ computed with m samples.

Theorem 4.1 (Hoeffding Inequality) *Let $\delta \in (0,1)$ be a parameter, vectors $\boldsymbol{p}, \boldsymbol{p}_m$ and $\boldsymbol{v}$ defined as above. Then with probability at least $1-\delta$,*

$$\left|\boldsymbol{p}^\top \boldsymbol{v} - \boldsymbol{p}_m^\top \boldsymbol{v}\right| \leq \|\boldsymbol{v}\|_\infty \cdot \sqrt{2m^{-1}\log(2\delta^{-1})}.$$

Theorem 4.2 (Bernstein Inequality) *Let $\delta \in (0,1)$ be a parameter, vectors $\boldsymbol{p}, \boldsymbol{p}_m$ and $\boldsymbol{v}$ defined as in Theorem 4.1. Then with probability at least $1-\delta$*

$$\left|\boldsymbol{p}^\top \boldsymbol{v} - \boldsymbol{p}_m^\top \boldsymbol{v}\right| \leq \sqrt{2m^{-1} \operatorname*{Var}_{s' \sim \boldsymbol{p}}(\boldsymbol{v}(s')) \cdot \log(2\delta^{-1})} + (2/3)m^{-1}\|\boldsymbol{v}\|_\infty \cdot \log(2\delta^{-1}),$$

where $\operatorname*{Var}_{s' \sim \boldsymbol{p}}(\boldsymbol{v}(s')) = \boldsymbol{p}^\top \boldsymbol{v}^2 - (\boldsymbol{p}^\top \boldsymbol{v})^2$.

Now we can present the derivation of Lemma 5.1.

Proof By Theorem 4.2, for a singular state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, with probability at least $1 - \delta/(4|\mathcal{S}||\mathcal{A}|)$,

$$\begin{aligned}
\left|\tilde{\boldsymbol{w}}(s,a) - \boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}\right| &\leq \sqrt{2\boldsymbol{\sigma}_{\boldsymbol{v}^{(0)}} \cdot m_1^{-1} \cdot \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})} + 2 \cdot (3m_1)^{-1} \cdot \|\boldsymbol{v}^{(0)}\|_\infty \cdot \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1}) && (25)\\
&= \sqrt{2\boldsymbol{\sigma}_{\boldsymbol{v}^{(0)}} \cdot m_1^{-1} \cdot L} + 2 \cdot (3m_1)^{-1} \cdot \|\boldsymbol{v}^{(0)}\|_\infty \cdot L && (26)
\end{aligned}$$

By the union bound, this inequality holds for all (s,a) pairs with probability at least $1 - \delta/4$. This is the first inequality of the lemma.

Similarly, by Theorem 4.1 and a union bound over all (s,a) pairs, with probability at least $1-\delta/4$, for every (s,a), we have

$$\begin{aligned}
\left|\tilde{\boldsymbol{w}}(s,a) - \boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}\right| &\leq \|\boldsymbol{v}^{(0)}\|_\infty \cdot \sqrt{2m_1^{-1}\log(8|\mathcal{S}||\mathcal{A}|\delta^{-1})} && (27)\\
&= \|\boldsymbol{v}^{(0)}\|_\infty \cdot \sqrt{2m_1^{-1}L} && (28)
\end{aligned}$$

Then, by Eqn. 28,

$$\begin{aligned}
\left|\tilde{\boldsymbol{w}}(s,a)^2 - (\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)})^2\right| &= |\tilde{\boldsymbol{w}}(s,a) + \boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}| \cdot |\tilde{\boldsymbol{w}}(s,a) - \boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}| && (29)\\
&\leq \left[2\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)} + \|\boldsymbol{v}^{(0)}\|_\infty \cdot \sqrt{2m_1^{-1}L}\right] \cdot \left[\|\boldsymbol{v}^{(0)}\|_\infty \cdot \sqrt{2m_1^{-1}L}\right] && (30)\\
&\leq 2(\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)}) \cdot \|\boldsymbol{v}^{(0)}\|_\infty \cdot \sqrt{2m_1^{-1}L} + \|\boldsymbol{v}^{(0)}\|_\infty^2 \cdot 2m_1^{-1}L. && (31)
\end{aligned}$$

Since $\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)} \leq \|\boldsymbol{v}^{(0)}\|_\infty$, we obtain

$$\left|\tilde{\boldsymbol{w}}(s,a)^2 - (\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)})^2\right| \leq 2\|\boldsymbol{v}^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1}L} + \|\boldsymbol{v}^{(0)}\|_\infty^2 \cdot 2m_1^{-1}L, \tag{32}$$

Provided that $m_1 \geq 2L$, $2m_1^{-1}L \leq \sqrt{2m_1^{-1}L} \leq 1$, so

$$\left|\tilde{\boldsymbol{w}}(s,a)^2 - (\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)})^2\right| \leq 3\|\boldsymbol{v}^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1}L} \tag{33}$$

Using Theorem 4.1 and taking a union bound over all (s,a) pairs, with probability at least $1-\delta/4$, for every (s,a), we have

$$\left| \frac{1}{m_1} \sum_{j=1}^{m_1} \boldsymbol{v}^2(s_{s,a}^{(j)}) - \boldsymbol{P}_{s,a}^\top \boldsymbol{v}^2 \right| \le \|\boldsymbol{v}^{(0)}\|_\infty^2 \cdot \sqrt{2L/m_1} \tag{34}$$

By a union bound, we obtain, with probability at least $1-\delta/2$,

$$\begin{aligned} \left|\widehat{\boldsymbol{\sigma}}(s,a) - \boldsymbol{\sigma}_{\boldsymbol{v}^{(0)}}(s,a)\right| &\le \left|\tilde{\boldsymbol{w}}(s,a)^2 - (\boldsymbol{P}_{s,a}^\top \boldsymbol{v}^{(0)})^2\right| + \left|m_1^{-1} \sum_{j=1}^{m_1} \boldsymbol{v}^2(s_{s,a}^{(j)}) - \boldsymbol{P}_{s,a}^\top \boldsymbol{v}^2\right| \\ &\le 4\|\boldsymbol{v}^{(0)}\|_\infty^2 \cdot \sqrt{2m_1^{-1}L}. \end{aligned} \tag{35}$$

By a union bound, with probability at least $1-\delta$, both (26) and (35) hold, concluding the proof.

5 Conclusion

The algorithm provided by the authors computes an ϵ-optimal policy with probability 1-δ where both the time spent, and the number of samples taken, are upper bounded by:

$$O\left[\frac{|S||A|}{(1-\gamma)^3\epsilon^2} \log\left(\frac{|S||A|}{(1-\gamma)\delta\epsilon}\right) \log\left(\frac{1}{(1-\gamma)\epsilon}\right)\right] \tag{36}$$

This bound had previously only been achieved for sample complexity by making assumptions on ϵ. The authors here bound both sample and time complexity while allowing for any fixed ϵ. For an arbitrary ϵ, this improves upon the previous best-known bounds by a factor of $(1-\delta)^{-1}$, which is a significant improvement. The researchers also extend the method to computing ϵ-optimal policies for finite-time horizon MDPs with a generative model, and provide a nearly matching sample complexity lower bound. This is important as it helps to further our understanding of the underlying principles of reinforcement learning and influence future research for more efficient RL.

References

[AMK13] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J. Kappen. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349, 2013.

[SWW$^+$18] Aaron Sidford, Mengdi Wang, Xian Wu, Lin Yang, and Yinyu Ye. Near-optimal time and sample complexities for solving markov decision processes with a generative model. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[SWWY18] Aaron Sidford, Mengdi Wang, Xian Wu, and Yinyu Ye. Variance reduced value iteration and faster algorithms for solving markov decision processes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 29, pages 770–787. SIAM, 2018.