Abstract

We first show how policy gradient techniques can have very long running times and connect this issue with a fundamental intertwining of "exploration" and "exploitation". Coupled with the known drawbacks of approximate value function methods (no guaranteed policy improvement), we find all standard approaches to be inadequate. Drawing upon the strengths of each approach we propose a new reinforcement learning algorithm with new performance guarantees. This new algorithm: (1) can incorporate exploration (2) is guaranteed to improve a performance metric (3) is guaranteed to terminate in a 'small' number of timesteps and (4) can use function approximation. Upon termination the policy returned is "approximately optimal".

CONSERVATIVE POLICY ITERATION

1. Introduction

This paper presents a new algorithm for solving reinforcement learning problems in the MDP setting.

todo:

(1) Talk about how we _must_ have function approximation.
(2) Talk about notions of optimality given function approximation.
(3) Name algorithm.
(4) Fit proofs into itsy bitsy columns

In recent years, there has been a growing interest in direct policy-gradient methods for approximate planning in large Markov decision problems (MDP) to circumvent the problems with approximate value function methods. Approximate value function methods try to exploit the power of dynamic programming but are hindered due to the lack of performance improvement guarantees. We show that the lack of explicit exploration in gradient methods lead to unreasonably high variance estimates. We use ideas from both methods to motivate a new algorithm that converges to a 'good' policy in polynomial time, where 'good' is defined with respect to the degree of approximation being used.

2. Preliminaries

A reinforcement learning problem is a finite Markov decision process (MDP) with rewards which is a tuple $(S, D, A, \mathcal{R}, P)$ where: S is finite set of states, D is the starting state distribution, A is a finite set of actions, $\mathcal{R}$ is a reward function $\mathcal{R} : S \times A \to [0, R]$, and P is the transition model in which p^a_{ij} is the transition probability from state i to j under action a.

The agent's decision making procedure is characterized by a stochastic policy $\pi(a|s)$, which is the probability of taking action a in state s. Although in many situations, the goal of the agent is to maximize the average reward, we only consider the case of maximizing the γ-discounted average reward from the starting state distribution D, which has a similar solution for processes that 'mix' on a reasonable timescale [?]. It is convenient to define a distribution over future states, with respect to some arbitrary start-state distribution μ, in the following way:

$$d(s|\pi, \mu, \gamma) \equiv (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|\pi, \mu) \tag{2.1}$$

This is analogous to the stationary distribution in the infinite time setting, since as $\gamma \to 1$, $d(s|\pi, \mu, \gamma)$ tends to the stationary distribution. For brevity, we will sometimes write this as $d(s|\pi)$ when γ and μ are clear from context.

Let $\delta(s)$ be the distribution over states which concentrates all mass on state s. We define the value function as the expected reward under $\pi(a|s)d(s|\pi, \delta(s))$, or

$$J_\pi(s) \equiv E_{a,s' \sim \pi(a|s')d(s'|\pi,\delta(s),\gamma)} \mathcal{R}(s', a) \tag{2.2}$$

which is equivalent to usual definition of $J_\pi(s) \equiv (1-\gamma) E\{\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)|\pi, s_0 = s\}$, except we are using *normalized* values. The state-action value is defined as: $Q_\pi(s,a) \equiv (1-\gamma)\mathcal{R}(s,a) + \gamma \sum_{s'} p^a_{ss'} J_\pi(s')$, and the advantage is defined as $A_\pi(s,a) \equiv Q_\pi(s,a) - J_\pi(s)$. We will sometimes suppress the π subscript when the policy is clear from context.

The goal of the agent is to maximize the discounted reward from the start state distribution D,

$$\eta(\pi) \equiv E_{s \sim D}\{J_\pi(s)\} .$$

A well known result is that a solution exists which simultaneously maximizes $J_\pi(s)$ for all states.

3. The Problems with Current Methods

We now examine in more detail the problems with approximate value function methods and policy gradient methods. There are three questions we desire answers to:

(1) Is there some performance measure that is guaranteed to improve at every step?
(2) How difficult is it to verify if a particular update will improve this measure?
(3) After a reasonable number of policy updates, what performance level is obtained?

We now argue that both greedy dynamic programming and gradient methods give unsatisfactory answers to these questions. Note that one question we did not ask is "What is the quality of the asymptotic policy?". We are only interested in policies that we can find in a reasonable amount of time. Understanding the problems with these current methods will give insight into our new algorithm, which addresses these three questions.

3.1 The Weakness of Approximate Value Function Methods

Approximate value function methods suffer from a paucity of theoretical results on the performance of a policy based on the approximate values. This leads to weak answers to all three questions.

Let us consider some function approximator $f(s)$ with the l_∞-error:

$$\varepsilon = \max_s |f(s) - J_\pi(s)|$$

where π is some policy. Let J_g the value of a greedy policy based on this approximation. We have the following guarantee (see [?]) for all states s:

$$J_g(s) \geq J_\pi(s) - \frac{2\gamma\varepsilon}{1-\gamma}. \tag{3.1}$$

In other words, the performance is guaranteed to not decrease by more than $\frac{2\gamma\varepsilon}{1-\gamma}$. Question 2 is not applicable since these methods don't guarantee improvement and a performance measure to check isn't well defined.

For approximate methods, the time required to obtain some performance level is not well understood. At best, some asymptotic results exist (see [?]) . In contrast, exact value function methods have bounds showing how fast the values converge to optimal (see [?]).

3.2 The Convergence Problem for Policy Gradients

For policy gradient techniques, Question 1 has the appealing answer that the performance measure of interest is guaranteed to improve under gradient descent.

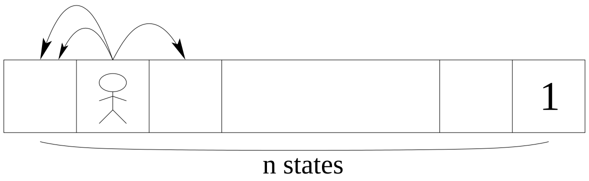

Figure 3.1. Example 1: Two actions move agent to the left and one actions moves agent to the right.

We now address question 2 by examining the situations in which estimating the gradient direction will be difficult. We view the lack of exploration in gradient methods as translating into requiring a very large number of timesteps in order to accurately estimate the gradient *direction*.

Consider the simple MDP shown in Figure 3.1 (adapted from [?]). Under a policy that gives equal probability to all actions, the expected time to the goal from the left most state is $3(2^n - n - 1)$, and with $n = 50$, the expected time to the goal is about 10^{15}. In general, many MDPs behave in the following manner: random actions are more likely than not to increase the distance to the goal state. For these classes of problems (see [?]), the expected goal-reaching time using undirected exploration, *ie* random walk exploration, is exponential in the size of the state space. Thus, any on-policy method will have to run for at least this long before any gradient estimate can occur. In online value function methods, this problem is seen as a lack of exploration.

Any sensible estimate of the gradient without reaching the goal state would be zero, and to obtain a non-zero estimate would require exponential time with on-policy samples. Importance sampling methods do exist (see [?, ?]), but these approaches lead to exponentially large importance weights or do not fix the need for an exponentially long exploration.

Note that a zero estimate is a rather accurate estimate of the gradient in terms of magnitude, but this provides no information about direction. The direction of the gradient is the crucial quantity of interest since we must step in the "right" direction in order to improve our policy. Some analysis has been done studying the deviation of gradient estimates [?]. However, this analysis focused on obtaining the policy gradient accurately (within some ε tolerance) rather than on obtaining the correct sign. This is a crucial difference because the magnitude of the gradient can be very small when the policy is far from optimal.

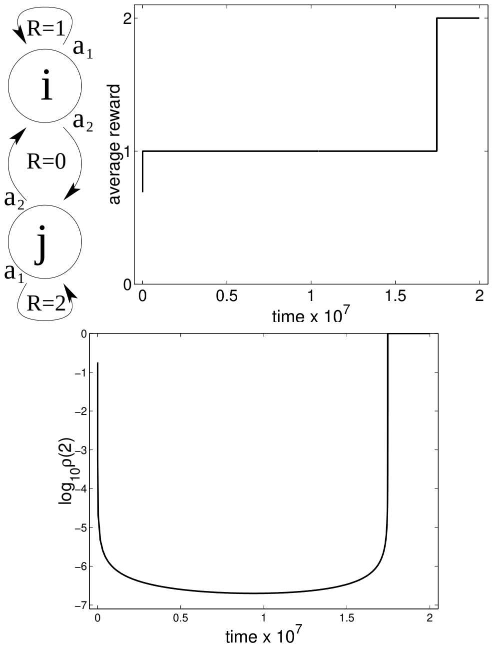

Figure 3.2. Example 2: A) See text. B) The average reward vs. time (on a 10^7 scale) of a policy under standard gradient descent in the limit of an infinitesimally small learning rate (initial conditions stated in text). C) The stationary probability of state j in B vs. time (on a 10^7 scale).

Let us give an additional example demonstrating the problems for a simple two state MDP as shown in Figure 3.2A, which uses the common Gibbs table-lookup distribution, $\tilde{\Pi}_G = \{\pi_\theta : \pi(a|s) \propto \exp(\theta_{sa})\}$. Increasing the chance of a self-loop at i decreases the stationary probability of j, which hinders the learning at state j. Under an initial policy that has the stationary distribution $\rho(i) = .8$ and $\rho(j) = .2$ (using $\pi(a_1|i) = .8$ and $\pi(a_1|j) = .9$), learning at state i reduces the learning at state j leading to an an extremely flat plateau of improvement at 1 average reward shown in Figure 3.2B. Figure 3.2C shows that this problem is so severe that $\rho(j)$ drops as low as 10^{-7} from it's initial probability of .2. As in example 1, to obtain a nonzero estimate of the gradient it is necessary to visit state j. The situation could be even worse with a few extra states in a chain as in figure 3.1.

Although in the limit a good policy might be found, these results do not bode well for the answer to question 3, which is concerned with how fast we can get there. These results suggest that in any reasonable number of steps, a gradient method could end up being trapped at plateaus where estimating the gradient direction is essentially impossible. The asymptotic results of standard stochastic gradient ascent techniques show that as the number of sample go to ∞ a local optima is found. Unfortunately, these 'asymptopia' results hide the sample complexity issues outlined here. Answering question 3 is crucial to understand how well gradient methods perform, and (to our knowledge) no such analysis exists.

4. The big picture

The fundamental problem with policy gradients is that η, which is what we ultimately seek to optimize, is insensitive to policy improvement at unlikely states though policy improvement at these unlikely states might be necessary in order for the agent to achieve near optimal payoff. Note that in example 1, policy improvement at state 49 will have little impact on the current performance, since the agent hardly ever reaches state 49 anyway. However, policy improvement at these unlikely states can ultimately lead to drastic policy improvement (as in figure 3.1).

We desire an alternative performance measure that does not down weight advantages at unlikely states or unlikely actions. A natural candidate for a performance measure is to weight the improvement from all states more uniformly (rather than by D), such as

$$\omega(\pi) = E_{s\sim\mu}\{J_\pi(s)\},$$

where μ is some 'exploratory' starting state distribution. To this end, ***we assume we have the ability to use an arbitrary start-state distribution*** μ. Note that for the special case when $\mu = D$ then $\omega = \eta$.

Any optimal policy simultaneously maximizes both ω and η. Unfortunately, the policy that maximizes ω within some restricted class of policies may have poor performance according to η. In general, the quality of the optima of ω is related to the the quality of the optima under η.

Greedy policy iteration updates the policy to some π' based on some approximate state-action values. Instead, let us consider the following more conservative update rule:

$$\pi_{\text{new}}(a|s) = (1-\alpha)\pi(a|s) + \alpha\pi'(a|s)\,, \tag{4.1}$$

for some π' and $\alpha \in [0,1]$. To guarantee improvement with $\alpha = 1$, π' must choose a better action at every state, or else we could suffer the penalty shown in equation 3.1.

A more reasonable situation is one in which we are able to improve the policy with some $\alpha > 0$ using a π' that chooses better actions at most but not all states. Let us define the *policy advantage* Π of some policy π' with respect to the policy π and the start-state distribution

μ to be:

$$\mathbb{A}_{\pi,\mu}(\pi') \equiv E_{s\sim d(s|\pi,\mu)}\left[\sum_a \pi'(a|s)A_\pi(s,a)\right].$$

Intuitively, this measures the expected advantage of π' over π, with respect to the set of states visited under π. Note that a policy found by one step of policy improvement will maximize the policy advantage.

Theorem 4.1. *Let $\mathbb{A}$ be the policy advantage of π' with respect to π and μ. If $\mathbb{A} \geq 0$, then using $\alpha = \frac{(1-\gamma)\mathbb{A}}{4R}$ with the update rule*

$$\pi_{new}(a|s) = (1-\alpha)\pi(a|s) + \alpha\pi'(a|s)$$

will guarantee the following policy improvement:

$$\omega(\pi_{new}) - \omega(\pi) \geq \frac{\mathbb{A}^2}{8R}.$$

Hence, the greater the policy advantage the greater the guaranteed increase in performance.

We now address question 3, by first addressing how fast we converge to some policy then bounding the quality of this policy. Naively, we expect our ability to obtain policies with large advantages to effect the speed of improvement and the quality of the final policy. Thus, the difficulty of finding an optimal policy will be related to the difficulty of solving a supervised learning problem: "What is a policy π' with a large policy advantage on policy π?" We assume access to some algorithm which solves this problem. This assumption allows us to analyze the difficulty of finding a near-optimal policy given the solution to a supervised learning problem which *may* incorporate function approximation. Let us call this supervised learning algorithm $BlackBoz(\pi,\mu)$. We assume access to an ε-good $BlackBox(\pi,\mu)$ defined as:

Definition 4.2. $BlackBox_\varepsilon(\pi,\mu)$ is a function of a policy π and start-state distributions μ which returns a policy π' such that $\mathbb{A}_{\pi,\mu}(\pi') \geq \mathrm{OPT}(\mathbb{A}_{\pi,\mu}) - \varepsilon$, where $\mathrm{OPT}(\mathbb{A}_{\pi,\mu}) = \max_{\pi'}\mathbb{A}_{\pi,\mu}(\pi')$.

We drop the μ dependence when clear. The "breaking point" of our supervised learning algorithm is reached when it is no longer guaranteed to return a policy with a positive policy advantage (i.e. when $\mathrm{OPT}(\mathbb{A}_{\pi,\mu}) - \varepsilon < 0$).

For simplicity, we assume we have knowledge of ε. A crude outline of the algorithm we suggest is:

(1) Call $BlackBox_\varepsilon(\pi,\mu)$ to obtain some π'
(2) Use $O(\frac{R^2}{\varepsilon^2}\log\frac{R^2}{\delta\varepsilon^2})$ Monte Carlo estimates to obtain an estimate $\hat{\mathbb{A}}$ of $\mathbb{A}_{\pi,\mu}(\pi')$.
(3) If $\hat{\mathbb{A}} < \frac{2\varepsilon}{3}$, STOP and return π.
(4) If $\hat{\mathbb{A}} \geq \frac{2\varepsilon}{3}$, then update policy π according to equation 4.1 using $\frac{(1-\gamma)(\hat{\mathbb{A}}-\frac{\varepsilon}{3})}{4R}$ and return to step 1.

The following theorem shows that in polynomial time, the algorithm finds a policy π that is 'close' to the break-point of the black box.

Theorem 4.3. *With probability at least $1-\delta$, algorithm ** improves ω with every policy update and the algorithm will cease in at most $72\frac{R^2}{\varepsilon^2}$ calls to $BlackBox_\varepsilon(\pi,\mu)$, and returns a policy π such that $OPT(\mathbb{A}_\pi) < 2\varepsilon$.*

To complete the answer to question 3, we need to address the quality of the policy found by this algorithm.

Theorem 4.4. *(simple version) Assume we have some π such that $OPT(\mathbb{A}_\pi) < \varepsilon$. Let π_* be an optimal policy and let $d_0 = \min_s d(s|\pi,\mu)$. Then*

$$\max_s |J_{\tilde{\pi}}(s) - J_{\pi_*}(s)| \leq \frac{\varepsilon}{(1-\gamma)d_0}.$$

This bound elucidates the problem of using the given start-state distribution, D, instead of a more uniform distribution. Intuitively, the maximal policy advantage with respect to π can be small even if there exist large advantages ($A_\pi(s,a)$). The presence of large advantages, even at states not currently visited, leaves open the possibility that π is far from optimal. Exploration is necessary to improve the policy at these unlikely states. The use of a uniform metric will bound $d_0 \geq \frac{1-\gamma}{\text{size of state space}}$.

We now discuss and prove these theorems in more detail.

5. Policy Improvement

We now prove theorem 4.1. First, let us examine, the policy change for small α. The gradient of a parameterized policy is

$$\nabla\omega = \sum_{s,a} d(s|\pi,\mu)\nabla\pi(a|s)Q_\pi(s,a),$$

(see [?]), which is equivalent to $E_{s\sim d(s|\pi,\mu)}\sum_a \nabla\pi(a|s)A_\pi(s,a)$. Therefore,

$$\begin{aligned}\frac{\partial\omega}{\partial\alpha}|_{\alpha=0} &= \frac{\sum_{s,a} d(s|\pi,\mu)(\pi'(a|s)-\pi(a|s))A_\pi(s,a)}{1-\gamma}\\ &= \frac{1}{1-\gamma}\sum_{s,a} d(s|\pi,\mu)\pi'(a|s)A_\pi(s,a)\end{aligned}$$

since $\sum_a \pi(a|s)A_\pi(s,a) = 0$. The change in ω is:

$$\Delta\omega = \frac{\alpha}{1-\gamma}\mathbb{A}_{\pi,\mu}(\pi') + O(\alpha^2). \tag{5.1}$$

Hence, for sufficiently small α, policy improvement can occur if the policy advantage is positive, and at the other extreme of $\alpha = 1$, significant degradation could occur. We now connect these two regimes.

The following lemma will be useful throughout:

Lemma 5.1. *For any policy $\tilde{\pi}$ and start-state distribution μ,*

$$\omega_{\tilde{\pi}}(s) - \omega_{\pi}(s) = \frac{1}{1-\gamma} E_{s,a\sim d(s|\tilde{\pi},\mu)\tilde{\pi}}\{A_\pi(s,a)\}$$

Proof. Let $d_t(s)$ be probability that the state at time t is s while following $\tilde{\pi}$ under the start-state distribution μ. We have

$$\begin{aligned}
& \omega(\tilde{\pi}) \\
= & (1-\gamma)\sum_{t=0}^{\infty}\gamma^t E_{s,a\sim d_t\tilde{\pi}}\{\mathcal{R}(s,a)\} \\
= & \sum_{t=0}^{\infty}\gamma^t E_{s,a\sim d_t\tilde{\pi}}\{(1-\gamma)\mathcal{R}(s,a) - J_\pi(s) + J_\pi(s)\} \\
= & E_{s\sim d_0}\{J_\pi(s)\} \\
& +\sum_{t=0}^{\infty}\gamma^t E_{s',s,a\sim d_{t+1}d_t\tilde{\pi}}\{(1-\gamma)\mathcal{R}(s,a) + \gamma J_\pi(s') - J_\pi(s)\} \\
= & \omega(\pi) + \sum_{t=0}^{\infty}\gamma^t E_{s,a\sim d_t\tilde{\pi}}\{A_\pi(s,a)\} \\
= & \omega(\pi) + \frac{1}{1-\gamma}E_{s,a\sim d(s|\tilde{\pi},\delta(s))\tilde{\pi}}\{A_\pi(s,a)\}\,.
\end{aligned}$$

□

The following theorem bounds the policy change for all α.

Theorem 5.2. *Let $\mathbb{A}$ be the policy advantage of π' with respect to the policy π and the distribution μ and let $\varepsilon = \max_s|\sum_a \pi'(a|s)A_\pi(s,a)|$. For the update rule 4.1 and for all $\alpha \in [0,1]$:*

$$\omega(\pi_{new}) - \omega(\pi) \geq \frac{\alpha}{1-\gamma}(\mathbb{A} - \frac{2\alpha\gamma\varepsilon}{1-\gamma(1-\alpha)})\,.$$

Before we state the proof, let us examine this result. The quantity ε is the largest average advantage, under π'. The first term is analogous to the first order increase specified in theorem 5.1, and the second term is a penalty term. Note that if $\gamma = 0$ then there is no penalty, so using $\alpha = 1$ will not result in policy degradation. Also note that if $\alpha = 1$, the bound reduces to

$$\omega(\pi_{\text{new}}) - \omega(\pi) \geq \frac{\mathbb{A}}{1-\gamma} - \frac{2\gamma\varepsilon}{1-\gamma}\,,$$

and the penalty term is identical to that of greedy dynamic programming (see equation 3.1).

The intuition for the proof is that α determines the probability of choosing an action from π'. If the current state distribution is $d(s|\pi,\mu)$ when an action from π' is chosen then the performance change from this deviant action will be proportional to the policy advantage. However, if the deviant action is chosen when the state distribution is different, the bound assumes the worst case situation where the agent chooses the action with the lowest advantage.

Proof. The expectated advatage with respect to π_{new} is,

$$\begin{aligned}
& E_{a\sim\pi_{\text{new}}(a|s)}\{A_\pi(s,a)\} \\
= & \sum_a((1-\alpha)\pi(a|s) + \alpha\pi'(a|s))A_\pi(s,a) \\
= & \alpha\sum \pi'(a|s)A_\pi(s,a) \\
= & \alpha E_{a\sim\pi'(a|s)}\{A_\pi(s,a)\}\,.
\end{aligned}$$

For any timestep, the probability that we choose an action according to π' is α. Let c_t be the random variable indicating the number of actions chosen from π' before time t. In particular, while no action has been chosen from π' we have $c_t = 0$ and after one action has been chosen $c_t = 1$. Hence, $\Pr(c_t = 0) = (1-\alpha)^t$ and $\Pr(c_t \geq 1) = 1-(1-\alpha)^t$. For any π', let $d_t(s|\pi')$ be the probability that the state at time t is s while following π' under the start-state distribution μ. It follows that for all timesteps t:

$$\begin{aligned}
& E_{a,s\sim\pi_{\text{new}}(a|s)d_t(s|\pi_{\text{new}})}A_\pi(s,a) \\
= & \alpha E_{a,s\sim\pi'(a|s)d_t(s|\pi_{\text{new}})}A_\pi(s,a) \\
= & \alpha\Pr(c_t = 0)E_{a,s\sim\pi'(a|s)d_t(s|\pi_{\text{new}},c_t=0)}A_\pi(s,a) \\
& +\alpha\Pr(c_t \geq 1)E_{a,s\sim\pi'(a|s)d_t(s|\pi_{\text{new}},c_t\geq 1)}A_\pi(s,a) \\
\geq & \alpha(1-\Pr(c_t \geq 1))E_{a,s\sim\pi'(a|s)d_t(s|\pi_{\text{new}},c_t=0)}A_\pi(s,a) \\
& -\alpha\Pr(c_t \geq 1)\varepsilon \\
\geq & \alpha E_{a,s\sim\pi'(a|s)d_t(s|\pi_{\text{new}},c_t=0)}A_\pi(s,a) - 2\alpha\Pr(c_t \geq 1)\varepsilon \\
= & \alpha E_{a,s\sim\pi'(a|s)d_t(s|\pi)}A_\pi(s,a) - 2\alpha(1-(1-\alpha)^t)\varepsilon
\end{aligned}$$

where we have used the definition of ε and $d_t(s|\pi_{\text{new}}, c_t = 0) = d_t(s|\pi)$.

By substitution and lemma 5.1, we have:

$$\begin{aligned}
& \omega(\pi_{\text{new}}) - \omega(\pi) \\
= \; & \frac{1}{1-\gamma} E_{a,s\sim\pi_{\text{new}}(a|s)d(s|\pi_{\text{new}})} A_\pi(s,a) \\
= \; & \sum_{t=0}^{\infty} \gamma^t E_{a,s\sim\pi_{\text{new}}(a|s)d_t(s|\pi_{\text{new}})} \{A_\pi(s,a)\} \\
\geq \; & \sum_{t=0}^{\infty} \gamma^t (\alpha \sum_{s,a} d_t(s|\pi)\pi'(a|s)A_\pi(s,a) - 2\alpha(1-(1-\alpha)^t)\varepsilon) \\
= \; & \frac{\alpha}{(1-\gamma)} E_{a,s\sim\pi'(a|s)d(s|\pi)} A_\pi(s,a) - 2\alpha\varepsilon \sum_{t=0}^{\infty} \gamma^t (1-(1-\alpha)^t) \\
= \; & \alpha \frac{\mathbb{A}_\pi(\pi')}{(1-\gamma)} - 2\alpha\varepsilon\left(\frac{1}{1-\gamma} - \frac{1}{1-\gamma(1-\alpha)}\right).
\end{aligned}$$

The result follows from simple algebra. $\square$

Appendix ** gives an example showing this bound is tight for all α.

Corollary 5.3. *Under the assumptions in the previous theorem and that R is the maximum possible reward, then*

$$\omega(\pi_{new}) - \omega(\pi) \geq \frac{\alpha}{1-\gamma}\left(\mathbb{A} - \alpha\frac{2R}{1-\gamma}\right).$$

Proof. The bound is a straightforward result of the previous theorem, using the fact that ε is bounded by R. $\square$

Theorem 1 follows by choosing the α that maximizes the lower bound in this corollary.

6. Conservative Policy Iteration

After our black box returns some policy π', we must address how to update the policy. If we knew the policy advantage of this policy was $\mathbb{A}$ and if $\mathbb{A} > 0$, then we could use $\alpha = \frac{(1-\gamma)\mathbb{A}}{4R}$.

Let us address how many samples are required to accurately determine $\mathbb{A}_\pi(\pi')$. We must specify a sampling model to estimate the policy advantage. It is straightforward to show that the policy advantage can be written as $\mathbb{A}_{\pi,\mu}(\pi') = E_{s\sim d(s|\pi,\mu)}\left[\sum_a (\pi'(a|s) - \pi(a|s))Q(s,a)\right]$ which is a more convenient form since we can directly estimate the state-action values.

Suppose that we can draw from the start-state distribution μ and have access to the MDP (in the sense that we can make it transition). Then, we can use monte carlo integration in order to estimate the value of $\mathbb{A}_{\pi,\mu}(\pi')$. This can be done by first drawing a state s from $d(s|\pi,\mu)$ with one draw $\mu(s)$ and $\Theta\left(\frac{1}{1-\gamma}\right)$ transitions in the MDP. Then, we can use importance sampling with a uniform proposal distribution over the actions to choose an action a. (We do not use π or π' as the importance distribution, since if either of these are deterministic the importance weights may be large.) Next, we run the MDP with policy π for $O\left(\frac{\ln\frac{R}{\epsilon}}{1-\gamma}\right)$ transitions in order to obtain an estimate $\hat{Q}(s,a)$ of $Q(s,a)$ with a bias of ϵ or less. The estimate of the policy advantage from the i-th sample will be $x_i = n_a\hat{Q}_i(s,a)(\pi'(a|s) - \pi(a|s))$ where n_a is the number of actions.

How many samples are required? Since $\hat{Q}_i \in [0,R]$, our samples satisfy $x_i \in [-n_aR, n_aR]$. Using Hoeffding's inequality for k independent identical random variables, we have:

$$\Pr\left(|\mathbb{A} - \hat{\mathbb{A}}| > \epsilon\right) \leq 2e^{-\frac{k\epsilon^2}{2n_a^2R^2}}, \tag{6.1}$$

where $\hat{\mathbb{A}} = \frac{1}{k}\sum_{i=1}^k x_i$. Hence, the number of samples required to achieve a fixed error rate is $O\left(\frac{n_a^2R^2}{\mathbb{A}^2}\right)$ and the number of MDP transitions required is $O\left(\frac{n_a^2R^2}{\mathbb{A}^2}\frac{1+\ln\frac{R}{\mathbb{A}}}{1-\gamma}\right)$. Consequently, if the policy advantage is large then few samples will be needed.

Now the proof of Algorithm ** is straightforward:

Proof. There are three statements, that must all hold with probability $1-\delta$, — ω improves at every step, the algorithms ceases in $\frac{72R^2}{\varepsilon^2}$ iterations, and the policy found has maximal policy advantage less than 2ε. Let us assume that every time we make a monte-carlo estimate of the policy advantage, our estimate $\hat{\mathbb{A}}$ of the policy advantage satisfies $|\mathbb{A} - \hat{\mathbb{A}}| < \frac{\varepsilon}{3}$. We will discuss the probability this assumption fails later.

The assumption implies that if we observe $\hat{\mathbb{A}} \geq \frac{2\varepsilon}{3}$, then $\mathbb{A} \geq \frac{\varepsilon}{3}$. By corollary 5.3, step 4 will guarantee improvement of $\frac{(\hat{\mathbb{A}} - \frac{\varepsilon}{3})^2}{8R} \geq \frac{\varepsilon^2}{72R}$ using $\alpha = \frac{(1-\gamma)(\hat{\mathbb{A}} - \frac{\varepsilon}{3})}{4R}$. Since $0 \leq \omega \leq R$, there are at most $\frac{72R^2}{\varepsilon^2}$ steps before ω becomes larger than R, so the algorithm must cease in this time. Our algorithm ceases and returns π when we observe that the policy advantage of π' satisfies $\hat{\mathbb{A}}_\pi(\pi') < \frac{2\varepsilon}{3}$ which implies $\mathbb{A}_\pi(\pi') < \varepsilon$. By definition of the black box, it follows that $\mathrm{OPT}(\mathbb{A}_\pi(\pi')) < 2\varepsilon$.

To obtain an error probability of less than δ, we must obtain enough samples such that $|\mathbb{A} - \hat{\mathbb{A}}| < \frac{\varepsilon}{3}$ for every loop of the algorithm, and there are at most $\frac{72R^2}{\varepsilon^2}$ loops. By the union bound and the inequality 6.1, we have $\frac{72R^2}{\varepsilon^2}2e^{-\frac{k\varepsilon^2}{2n_a^2R^2}} < \delta$, and so when $O\left(\frac{R^2}{\varepsilon^2}\log\frac{R^2}{\delta\varepsilon^2}\right)$

samples are used, the probability of failure is less than δ. $\square$

7. How Good is The Policy Found?

The speed with which the algorithm ceases does not depend on the start-state distribution used. However, we now show that the quality of the resulting policy could strongly depend on this distribution.

The reason is due to a fundamental *measure mismatch* in reinforcement learning, which we now analyse. The performance measure of interest, $\eta(\pi)$ (defined with respect to start-state distribution D), changes in proportion the policy advantage $\mathbb{A}_{\pi,D}(\pi')$ for small α when using our update rule (see equation 5.1). A resulting improvement of $\frac{\mathbb{A}^2}{8R}$ can be guaranteed if $\mathbb{A} \geq 0$. Recall that the policy advantage is just the expected advantage under the $d(s|\pi, D)\pi'(a|s)$. However, the difference between η and η^* is the average advantage under $d(s|\pi^*, D)\pi^*(a|s)$, where $\pi^*$is an optimal policy. Thus, if $\mathbb{A}_{\pi,D}(\pi')$ is small, this will not imply that η is close to optimal. For example, there could still be some large advantages where $d(s|\pi, D)$ is small and $d(s|\pi^*, D)$ is large. Also, with $\mathbb{A}_{\pi,D}(\pi')$ small, little improvement is guaranteed and even detecting if the policy advantage is positive ($\mathbb{A}_{\pi,D}(\pi') \geq 0$) will have large sample complexity. Intuitively, this is the situation where large advantages are at states with low probabilities of being visited under the current policy (as in examples 1 and 2).

We now quantify how far from optimal the resulting policy is using some start-state distribution μ. We use the following definitions for some distribution $d(s)$, $||d||_\infty = \max_s d(s)$ and $||d||_0 = \min_s d(s)$.

Theorem 7.1. *Assume we have some π such that OPT$(\mathbb{A}_{\pi,\mu}) < \varepsilon$. Let π^* be an optimal policy. Then for any μ',*

$$\begin{aligned}\omega_{\mu'}(\pi^*) - \omega_{\mu'}(\pi) &\leq \frac{\varepsilon}{(1-\gamma)} \frac{||d(s|\pi^*, \mu')||_\infty}{||d(s|\pi, \mu)||_0} \\ &\leq \frac{\varepsilon}{(1-\gamma)^2} \frac{||d(s|\pi^*, \mu')||_\infty}{||\mu||_0}.\end{aligned}$$

Proof. Let $||a|| = \sum_s |a(s)|$. By lemma 5.1,

$$\begin{aligned}&\omega_{\mu'}(\pi^*) - \omega_{\mu'}(\pi) \\ =\ & \frac{1}{1-\gamma} \sum_{s,a} d(s|\pi^*, \mu')\pi'(a|s)A_\pi(s,a) \\ \leq\ & \frac{1}{1-\gamma} \sum_{s} d(s|\pi^*, \mu') \max_a A_\pi(s,a) \\ \leq\ & \frac{1}{1-\gamma} ||d(s|\pi^*, \mu')||_\infty || \max_a A_\pi(s,a)||_1 .\end{aligned}$$

where $|| \max_a A_\pi(s,a)||_1 = \sum_s \max_a A_\pi(s,a)$. The optimal policy advantage is $\mathrm{OPT}(\mathbb{A}_{\pi,\mu}) = \sum_s d(s|\pi,\mu) \max_a A_\pi(s,a) \geq \sum_s ||d(s|\pi,\mu)||_0 \max_a A_\pi(s,a)$. Therefore, we have

$$\mathrm{OPT}(\mathbb{A}_{\pi,\mu}) \geq ||d(s|\pi,\mu)||_0 || \max_a A_\pi(s,a)||_1$$

Since we have $\varepsilon \geq \mathrm{OPT}(\mathbb{A}_{\pi,\mu})$, the first inequality follows. The second inequality is due to $||d(s|\pi,\mu)||_0 \leq (1-\gamma)||\mu||_0$. $\square$

It is important to notice that each $\frac{1}{1-\gamma}$ factor has an interpretation. In particular, one factor of $\frac{1}{1-\gamma}$ is intuitively due to the fact that approximation errors can be incurred over $\frac{1}{1-\gamma}$ timesteps and another factor of $\frac{1}{1-\gamma}$ is due to the inherent ununiformity of $d(s|\pi^*, \mu')$.

8. Notes

8.1 improvement along the way

Discuss using the mixed start state dist.

9. Discussion

(1) Practical variants — line searches
(2) representation
(3) decay??
(4) discuss sensible supervised learning methods??
(5) exploration!!

-0.12in0.02inReferences

10. Appendix B: Examples

The example for theorem **??** is easy.

FIG: I didn't have the time to make the fig, but I'll describe it. There are two states, i and j, and two actions, 1 and 2, at each state. At state i, action 1: $R(i,1) = 0$ and transition is $i \to i$ and action 2: $R(i,2) = R$ and transition is $i \to j$. At state j, action 1: $R(j,1) = 0$ and transition is $j \to j$ and action 2: $R(j,2) = -R$ and transition is $j \to j$. So only one action (1 at state i) can move the agent from one state to another. *-Sham*

We now show the bound in theorem 5.2 is tight. Consider the MDP shown in the figure. Let the start state be i. Let $\pi(1|i) = 1$ and $\pi(1|j) = 1$, so $d(i|\pi) = 1$. The advantage are then $A(i,1) = 0$, $A(i,2) = (1-\gamma)R$, $A(j,1) = 0$, and $A(j,2) = -(1-\gamma)R$. Let us choose

π' such that $\pi'(2|i) = 1$ and $\pi'(2|j) = 1$, so:

$$\begin{aligned}
\Pi_\pi(\pi') &= E_{s\sim d(s|\pi,\mu_e)}\{\sum_a \pi'(a|s)A_\pi(s,a)\} \\
&= 1*(1-\gamma)R + 0*(-(1-\gamma)R) \\
&= (1-\gamma)R\,,
\end{aligned}$$

so policy improvement is possible with π' since the policy advantage is positive. It is clear that $\omega(\pi) = 0$ and $\varepsilon = (1-\gamma)R$.

Now let us compute $\omega(\pi_{\text{new}})$. If the agent is at state 1, the immediate reward it receives is, on average,

αR and if the agent is at state 2, the immediate reward is $-\alpha R$ (based on π_{new}). It follows that

$$\begin{aligned}
\omega(\pi_{\text{new}}) - \omega(\pi) &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t r_t - 0 \\
&= \alpha(1-\gamma)\sum_{t=0}^{\infty}\gamma^t(P(s_t = i|t)R - P(s_t = j|t)R) \\
&= \alpha(1-\gamma)R\sum_{t=0}^{\infty}\gamma^t(1-2P(j|t)) \\
&= \alpha(1-\gamma)R\sum_{t=0}^{\infty}\gamma^t(1-2(1-(1-\alpha)^t)) \\
&= \alpha(1-\gamma)R\sum_{t=0}^{\infty}\gamma^t - 2\alpha(1-\gamma)R\sum_{t=0}^{\infty}\gamma^t(1-(1-\alpha)^t) \\
&= \alpha R - 2\gamma R\frac{\alpha^2}{1-\gamma(1-\alpha)} \\
&= \alpha\frac{\Pi}{1-\gamma} - \frac{\alpha^2}{1-\gamma(1-\alpha)}\frac{2\gamma\varepsilon}{1-\gamma}\,.
\end{aligned}$$

Hence, the bound is tight. Note that the we can improve by at least $\frac{\Pi^2}{8R} = \frac{(1-\gamma)^2 R}{8}$, with $\alpha = \frac{(1-\gamma)\Pi}{4R} = \frac{(1-\gamma)^2}{4}$.