MY NEW REINFORCEMENT LEARNING TECHNIQUE IS UNSTOPPABLE [, HTTP://WWW.MNFTIU.CC/MNFTIU.CC/FILING.HTML]

Abstract. (update me) Gradient methods fundamentally intertwine exploration and exploitation. For these techniques, determining if any gradient direction (in policy space) will improve the performance of the current policy can be arbitrarily difficult. To overcome these difficulties, an alternative performance measure which does not suffer from these problems is proposed and analyzed. This alternative performance measure has the same global optima as the average reward and can be optimized without decreasing the average reward given a sufficiently flexible function approximator.

1. Introduction

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. In contrast, direct policy-gradient methods seek to find a good policy π among some restricted class of policies, by following the gradient of the future reward. Gradient methods are essentially on-policy and so intertwine the fundamental exploration-exploitation tradeoff. We use ideas from both methods to motivate a new algorithm.

Earlier work has argued that random policies could lead to learning that is exponentially slow in the size of the state space [**?**]. For example, consider the simple MDP shown in Figure 1.1A (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}. Thus, any on-policy method will have to run for at least this long before any learning can occur. In online value function methods, this problem is seen as a lack of exploration. Active exploration is necessary to improve the policy at states that are not visited often in order to efficiently propagate rarely visited state value information to commonly visited states (see [**?**, **?**]). In gradient methods, we show that this lack of exploration leads to high variance estimates of the gradient.

Let us consider the complexity of the most uninformed initial policy, namely a random or *undirected* policy, in MDPs where the task is to reach some goal state as quickly as possible. Undirected policies are of interest since they are common initializations. If random actions are more likely to increase the distance to the goal state rather than decreasing it, a fundamental theorem by Whitehead [**?**, **?**]is now applicable:

Theorem 1.1. *(by Whitehead) In any finite homogeneous problem solving task, the expected time using undirected exploration (such as random walk exploration) required to identify an optimal policy is bounded below by an expression exponential*

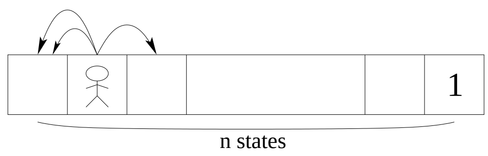

FIGURE 1.1. Example 1: Two actions move agent to the left and one actions moves agent to the right.

in the depth of the state space, if the number of actions leading away from the goal is larger than the the number of actions leading closer to the goal.

Can this be stated mathematically? -John

I imagine so, but I haven't got the original paper yet. I copied this excerpt out of one of Thrun's papers [**?**].

Sloppy statements should be avoided. -John

The proof of the theorem shows that the expected time to the goal state on the first trial is exponential in the size of the state space. We share the belief with Whitehead and others ([**?**]) that many real-world problems fulfill the condition that a random action will increase the expected goal distance. As a special case of this theorem, policy gradient methods will be intractable for these classes of problems.

Using an exact world model, the method of dynamic programming improves the policy at all states. In the method of policy iteration, the policy at unlikely states will exhibit policy improvement even though this improvement might have little impact on the current performance of the policy. 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. Unfortunately, with function approximation, greedy dynamic programming has only weak performance improvement guarantees.

The fundamental problem is that the current performance level of the policy is insensitive to states not likely to be visited under this policy. However, policy improvement at these unlikely states can ultimately lead to drastic policy improvement (as in figure 1.1).

First (in section 3), we will characterize the problems with current methods. The problems with greedy dynamic programming have already been well-studied [**?**]. We show that the problem with gradient methods is that estimating the gradient direction can be arbitrarily difficult to estimate when the policy could be arbitrarily far from a local optimum.

The remainder of the paper outlines a new algorithm and analyzes the performance of this algorithm. Our new algorithm is motived by gradient methods and dynamic programming. The important idea of gradient methods that we incorporate is that small changes to the policy are necessary in order to have reasonable performance improvement guarantees. The idea we borrow from value function methods is that it is necessary to improve the policy at states not visited often.

The outline of the paper is as follows:

- In section 2, the usual formulation of the reinforcement learning problem is stated.
- In section 3, the problems with greedy dynamic programming methods and gradient methods are exhibited in more detail.

- In section 4, we discuss the importance of using a new performance measure and analyse the difficulty of improving this measure.
- In section 5, we show that our algorithm can find an 'approximately' optimal policy in a polynomial number of steps.
- In section 6, we show that we can improve the alternative measure and the measure of interest and we give results on how the bounds are reasonable in 'real-world' problems (Lipshitz conditions). *etc. etc.*

2. Preliminaries

First, let us state the standard formulation of a reinforcement learning problem. A reinforcement learning problem is a finite Markov decision process (MDP) with rewards which is a tuple $(S, \mu, A, \mathcal{R}, P)$ where: S is finite set of states, μ 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_{ij}^{a} 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.

If we make the assumption that *every policy* π *is ergodic, ie* has a well-defined stationary distribution p_{stat}, then the *average reward* is $\sum_{s,a} p_{\text{stat}}(s)\pi(a|s)R(s,a)$. 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 μ. This approximation allows us to avoid mathematical difficulties of the average reward formulation while arriving at a similar solution for processes that "mix" on a reasonable timescale [**?**].

Given a fixed starting state distribution μ, a fixed (possibly nonstationary) policy π, and a fixed MDP, there exists a fixed distribution over states at time step t. Let this fixed distribution be

$$d_t(s) = \Pr(s_t = s|\pi, \mu)$$

Since we are concerned with the distribution over time, it is convenient to think of a distribution over future states defined 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. For brevity, we will sometimes write this as $d_\pi(s)$ when γ and μ are clear from context. As $\gamma \to 1$, $d_\pi(s)$ tends to the stationary distribution.

Let $\delta(s)$ be the distribution over states which concentrates all mass on s.

We define the value function as

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

and the state-action value as

$$Q_\pi(s, a) \equiv E_{s' \sim d(s'|\pi, \delta(s), \gamma)} \mathcal{R}(s', a) \tag{2.3}$$

where and s_t and a_t are random variables for the state and action at time t. The goal of the agent is to maximize the discounted reward from the start state distribution μ,

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

The advantage is defined by:

$$A_\pi(s,a) \equiv Q_\pi(s,a) - J_\pi(s)\,. \tag{2.4}$$

We will sometimes suppress the π subscript when the policy is clear from context.

A well known result is that a solution exists which simultaneously maximizes $J_\pi(s)$ for all states. Dynamic programming methods find this solution by using an algorithm which alternates between improving the value function estimates and the greedy policy based on these value function estimates. We refer to a greedy policy based on some $A_\pi(s,a)$ as a policy that only selects an action a at some state s if $a \in \mathrm{argmax}_{a'} A_\pi(s,a')$.

Gradient methods consider some parameterized class of policies and follow the gradient of η with respect to the policy parameters. The gradient of the discounted average reward from a starting state s_0 chosen according to μ is:

$$\nabla\eta(\pi) = \frac{1}{1-\gamma}\sum_{s,a} d_\pi(s) Q_\pi(s,a) \nabla\pi(a|s) \tag{2.5}$$

as shown in [**?**].

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. We also address these three questions with respect to our new algorithm.

3.1. **What is Wrong with 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 first address question 1. In exact dynamic programming methods, often the performance measure used is a maximum over the states between the current value and the optimal, $\max_s |J(s) - J_{\mathrm{opt}}(s)|$. This measure provably decreases in methods such as value iteration and policy iteration. Further, in policy iteration, there is the added guarantee that a greedy policy based on the exact state-action values is guaranteed to improve the policy starting form any state. These methods are not concerned with the particular starting distribution, since optimization leads to good performance from any starting distribution.

Unfortunately, these guarantees no longer hold for value function methods. 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 state 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* much. Approximate value iteration has even weaker theoretical guarantees. Question 2 is not applicable since these methods don't guarantee improvement and the a performance measure to check isn't well defined.

For exact methods, there are bounds showing how fast the values converge to optimal [?]. However, for approximate methods, answers to questions 3 are even less well understood. Typically, the algorithms used are not guaranteed to converge. This alone is not necessarily cause for concern if the final set of policies reached have good value. Let π_k be the sequence of policies generated by approximate policy iteration, with l_∞-error ε at each step. The only bound (to our knowledge) states that the final of set of policies is bounded satisfies:

$$\limsup_{k\to\infty} \max_s |J_{\pi_k}(s) - J^*(s)| \geq \frac{2\gamma\varepsilon}{(1-\gamma)^2}$$

though it is not clear how tight this bound is. Further, there are no guarantees as to how many steps it could take to for a policy get approximately this close to optimal, which is question 3's concern.

3.2. **What is Wrong with Policy Gradients?** Question 1 has the appealing answer that the performance measure of interest is guaranteed to improve if we follow the gradient. We now take some time to answer question 2 by examining the situations in which estimating the gradient direction will be difficult. In order to determine if an estimate of the gradient, with respect to the i-th parameter, will improve the policy, it is crucial to obtain the correct sign of $\nabla_i\eta$. We expect the difficulty of determining if $\nabla_i\eta > 0$ to be proportional to the ratio of the "noise" Σ to the "signal" $\nabla_i\eta$. Some analysis has been done studying the variance of gradient estimates [?] although this analysis focused on obtaining $\nabla_i\eta$ accurately, within some ε tolerance, rather than on obtaining the correct sign.

Fundamentally, the only process available to us is running the MDP of the reinforcement learning problem. Given certain assumptions (see section 12), it is reasonable to suppose that we can obtain samples in an on-policy manner, *ie* we can obtain samples of states -action pairs from $d(s|\pi,\mu)\pi(a|s)$ and we can obtain a nearly unbiased estimate $\hat{Q}(s,a)$ of $Q_\pi(s,a)$. Using importance sampling, the estimate of the gradient of $\nabla_i\eta$ is then $\hat{Q}(s,a)\frac{\nabla_i\pi}{\pi}$ (see equation 2.5).

Theorem 3.1. *Assume that samples s,a are drawn from the distribution $\nu(s,a) \equiv d(s|\pi,\mu)\pi(a|s)$ and that $\frac{\nabla_i\pi}{\pi}$ exists. For each sample, we are given an unbiased estimate $\hat{Q}(s,a)$ of $Q(s,a)$ with variance $\sigma^2(s,a)$. Let $\sigma_{min} = \min_{s,a}\sigma(s,a)$. Then the ratio of the variance, Σ, to the mean, μ, of the estimate $\hat{Q}(s,a)\frac{\nabla_i\pi}{\pi}$ satisfies:*

$$\frac{\Sigma^2}{\mu^2} \geq \frac{\sigma_{min}^2}{E_\nu A^2} - 1\,.$$

Before stating the proof, let us discuss the implications. Consider the case in which all the large advantages are at states that are visited infrequently, so $E_\nu A^2$ will be small. This theorem suggests that it may be difficult to determine if the gradient will improve the policy for *any parameterization*. We say 'suggests' since high signal to noise does not directly translate into a lower bound on the number of samples required. However, the presence of large advantages means the the policy could potentially be far from optimal (as in example 1). Notice for example 1 (and other in Whitehead's class [?]), this noise to signal ratio will be exponentially large. We view the lack of exploration in gradient methods as translating into giving high variance estimates for the gradient *direction* (not magnitude).

Proof. Since the $\hat{Q}$ are unbiased, the mean μ of the estimate $\hat{Q}(s,a)\frac{\nabla_i \pi}{\pi}$ is also unbiased. The variance of this estimate is:

$$\Sigma^2 = E\{\left(\frac{\nabla_i \pi}{\pi}\right)^2 \hat{Q}^2\} - \mu^2 \tag{3.2}$$

$$= E_{s,a\sim\nu}\{\int dP(\hat{Q})\hat{Q}^2 \left(\frac{\nabla_i \pi}{\pi}\right)^2\} - \mu^2 \tag{3.3}$$

$$= E_{s,a\sim\nu}\{(Q^2+\sigma^2)\left(\frac{\nabla_i \pi}{\pi}\right)^2\} - \mu^2 \,. \tag{3.4}$$

We can lower bound Σ as follows:

$$\begin{aligned} \Sigma^2 &\geq E_{s,a\sim\nu}\{\sigma^2 \left(\frac{\nabla_i \pi}{\pi}\right)^2\} - \mu^2 \\ &\geq \sigma^2_{\min} E_{s,a\sim\nu}\left(\frac{\nabla_i \pi}{\pi}\right)^2 - \mu^2 \,. \end{aligned}$$

Note that since $\sum_a \nabla\pi(a|s) = 0$ then $E_\nu\left[Q_\pi \frac{\nabla_i \pi}{\pi}\right] = E_\nu\left[A_\pi \frac{\nabla_i \pi}{\pi}\right]$. It follows that $\frac{\Sigma^2}{\mu^2} \geq \frac{\sigma^2_{\min} E_\nu\left(\frac{\nabla_i \pi}{\pi}\right)^2}{(E_\nu A \frac{\nabla_i \pi}{\pi})^2} - 1$. Note that for vectors x and y in state-action space, $E_\nu(xy)$ is an inner product rule. By the Cauchy-Schwartz inequality, $(E_\nu A\frac{\nabla_i \pi}{\pi})^2 \leq E_\nu A^2 E_\nu(\frac{\nabla_i \pi}{\pi})^2$, and the result follows. $\square$

An important point is that our sampling method for gradient estimation is on-policy. Another possibility is to use importance sampling to try and reduce the variance. Such sampling methods have been formulated [?, ?]. However, the fundamental problem is that there could be a significant contribution to the expectation of interest from a state with an exponentially low probability of being visited (as in example 1). This will lead to unreasonably large importance weights, which does not bode well. We do not pursue this direction here.

Let us give an additional example demonstrating the problems for a simple two state MDP as shown in Figure ??B. Our example will use 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 $\rho(j) = .2$, 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 ??C (see figure text). Figure ??D shows that this problem is so severe that $\rho(j)$ drops as low as 10^{-7} from it's initial probability of .2, which makes sampling the gradient difficult. Figure ??E shows where this occurs and gives a clearer picture (see figure text). This example is remarkably discouraging for policy gradient approaches. We can easily make it even worse by using a few extra states in a chain as in figure 1.1. This problem also demonstrates that the mixing times are not necessarily relevant to having high variance gradient estimates, since the policy mixes very quickly when $\rho(j) = 10^{-7}$.

One question immediately arises: are we suffering from numerical error problems? The answer is no, as can be shown by a careful (and somewhat hairy) analysis of reinforcement learning with the Gibbs distribution. The analysis is given in section 9.

Although in the limit a good policy might be found (see appendix), 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 reasonable number of steps a gradient method could end up being trapped at plateaus where estimating the gradient direction has high variance. The asymptotic results of standard stochastic gradient ascent techniques show that as the number of sample go to ∞ a local optima is found, though this hides the variance trap issue that we have outlined. Answering question 3 is crucial to understand how well gradient methods perform, and (to our knowledge) no such analysis exists.

4. Optimizing a Different Performance Measure

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. We desire an alternative performance measure that does not down weight advantages at unlikely states or unlikely actions. The performance measure we choose will require access to an exploration policy or simulator which can achieve a more "uniform" distribution over start-states. A natural candidate for a performance measure is to weight the improvement from all states more uniformly (rather than by μ), such as

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

where μ_e is some 'exploratory' starting *state-action* distribution. Note that for the special case when $\mu_e = \mu(s)$ then $\omega = \eta$. Under this performance measure, even if a state is not visited often (like the right most states in example 1), importance will be given to improving the future reward from this unlikely state if μ_e gives the state some weight. Note that if μ_e is non-vanishing at every state, then the policy that maximizes ω also simultaneously maximizes the future reward from every state, and so η is optimized.

Unfortunately, with a restricted class of parameterized policies, optimizing ω may result in poor performance according to η. In order to exploit the benefits of exploration, the class of policies we consider must allow us to improve the policy in different areas of the state space without ultimately preventing us from improving η. This tradeoff is dependent on the properties of the class of policies we consider.

This section will focus on addressing questions 1 and 2, and as special case, the results are applicable to gradient methods. In 4.1, we state general conditions for policy improvement in an incremental framework. In 4.2, we focus on the difficulty of determining if an update will improve our performance measure.

4.1. **A Method for Incremental improvement.** Let us consider an update rule in which we move the current policy π towards some new policy π'. More formally, consider the incremental update rule:

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

We later address how to choose a π' at each point in time. Note that for $\alpha \in [0,1]$ this a valid distribution, and that for negative α sufficiently close to 0, the policy is also a valid distribution.

Let us compare this to the gradient update rule,

$$\pi_{\mathrm{new}}(a|s) = \pi_\theta(a|s) + \frac{\partial \pi}{\partial \theta}\Delta\theta + O(\Delta\theta^2)$$

where the sign of $\Delta\theta$ determines if we move with or against the direction $\frac{\partial\pi}{\partial\theta}$ in policy space. The incremental update rule is similar to the gradient update rule when $\Delta\theta = \alpha$ is small and

$$\frac{\partial \pi}{\partial \theta} = \pi' - \pi\,. \tag{4.2}$$

Hence, we can view the choice of π' as giving a local search similar to the way a class of parameterized policies give a local search direction.

Consider using $\alpha = 1$, which makes the new policy π'. To guarantee improvement for the general case, with $\alpha = 1$, we have the stringent requirement that π' must choose a better action at every state, 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 using some $\alpha > 0$ with a π' that chooses better actions at most states but not all states.

Let us define the policy advantage Π of some policy π' with respect to the policy π and the arbitrary start-state distribution μ to be:

$$\Pi_{\pi,\mu}(\pi') = E_{a,s\sim\pi'(a|s),d(s|\pi,\mu)}\left[A_\pi(s,a)\right]\,.$$

Intuitively, this measures the expected advantage of π' over π, with respect to the set of states visited under π. Note that if a policy $\pi^{'}$ always greedily chooses the action with the largest advantage on the current policy, then iterative update with $\alpha = 1$ will never degrade the policy.

Theorem 4.1. *Let Π be the policy advantage of π' with respect to π and μ_e. Under the following update rule*

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

the performance improvement of ω is:

$$\Delta\omega = \frac{\alpha}{1-\gamma}\Pi + O(\alpha^2)\,. \tag{4.3}$$

Proof. Note that

$$\frac{\partial \pi_{\mathrm{new}}(a|s)}{\partial \alpha} = \pi'(a|s) - \pi(a|s)\,. \tag{4.4}$$

Since the probabilities are normalized $\sum_a \nabla\pi(a|s) = 0$ and so

$$\begin{aligned}\nabla\omega &= \frac{1}{1-\gamma}\sum_s d(s|\pi,\mu_e)\sum_a Q(s,a)\nabla\pi(a|s) \\ &= \frac{1}{1-\gamma}\sum_s d(s|\pi,\mu_e)\sum_a (A(s,a)+J(s))\nabla\pi(a|s) \\ &= \frac{1}{1-\gamma}\sum_s d(s|\pi,\mu_e)\sum_a A(s,a)\nabla\pi(a|s)\,.\end{aligned}$$

It follows that, to first order in α,

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

where we have used $\sum_a \pi(a|s)A(s,a) = 0$. □

Thus, to improve the policy it is necessary that π' have a non-zero policy advantage. Further, to first order, the performance will increase by the policy advantage over the horizon of $\frac{1}{1-\gamma}$.

A crude outline of the algorithm we suggest is:

(1) Choose a π' with a large policy advantage.
(2) Perform a line search to increase ω.

Now we discuss the importance of using a start-state distribution that gives weight to all states.

4.2. **The difficulty of improving a sub-optimal policy.** Let us now address how many samples are required to improve a sub-optimal policy. In order to do this, we must determine, with some certainty, the sign of $\Pi_{\pi,\mu_e}(\pi')$. We must now specify a sampling model to estimate the policy advantage. It is straightforward to show that the policy advantage can be written as:

$$\Pi_{\pi,\mu_e}(\pi') = E_{s\sim d(s|\pi,\mu_e)}\left[\sum_a (\pi'(a|s)-\pi(a|s))Q(s,a)\right]$$

which is a more convenient form since we can directly estimate the state-action values. As in section 3.2, we assume we can obtain independent samples of states from $d(s|\pi,\mu_e)$ under the exploratory start-state distribution μ_e (see section 12). To evaluate the sum over actions, we use importance sampling with a uniform proposal distribution over the actions, and assume we can then obtain an estimate of the respective state-action value. We do not use π or π' as the importance distribution, since if either of these are deterministic the importance weights may be large. The estimate of the policy advantage from one sample will then be $n_a\hat{Q}(s,a)(\pi'(a|s)-\pi(a|s))$ where n_a is the number of actions available in state s. Since the number actions is the same for all states s (according to our MDP definition), we suppress the dependence on s.

Theorem 4.2. *Let n_a be the number of possible actions at state s and let Π be the policy advantage of π' with respect to π and μ. Under the assumptions and sampling model of theorem 3.1, then the estimate $X = \frac{1}{k}\sum_{i=1}^{k} n_a \hat{Q}_i(a,s)(\pi'(a_i|s_i) - \pi(a_i|s_i))$ satisfies:*

$$\Pr(sign(X) \neq sign(\Pi)) \leq e^{-\frac{k\Pi^2}{2n_a^2 R^2}}$$

Proof. Since $\pi(s,a) \in [0,1]$ and $\hat{Q}_i \in [0,R]$, it follows that the random variable $n_a(\pi' - \pi)Q \in [-n_a R, n_a R]$. Using Hoeffding's inequality for k independent identical random variables, we have:

$$\Pr\left(|\Pi - X| > \epsilon\right) \leq e^{-\frac{k\epsilon^2}{2n_a^2 R^2}}\,.$$

Note that if $|\Pi - X| \leq |\Pi|$ then $\text{sign}(\Pi) = \text{sign}(X)$, so setting $\epsilon = \Pi$ gives the result. □

Hence, the number of samples required to achieve a fixed error rate is $O\left((\frac{n_a R}{\Pi})^2\right)$, so if the policy advantage is large then few samples will be needed.

If some policy π has large advantages, the policy advantage Π *can* still be small implying that a large number of samples may be required in order to guarantee the correct sign is chosen. The importance of this behavior is that if large advantages are present, the current policy could still be far from optimal. For example, an 'on-policy' weighting such as for policy gradient techniques can result in small policy advantages even when large advantages exist. This is the behavior observed in the examples 1.1 and 2.1. *Fixme -John*

Let us now formalize this. We measure the l_∞ error of the policy π by:

$$\Delta = \max_s |J(s) - J^*(s)|$$

which is the standard l_∞-measure used in dynamic programming (see [?]). Under this definition, η and ω could be at most Δ-far from optimal in a worst case scenario. Let us define the Bellman error of a policy π to be:

$$\varepsilon \equiv \max_s |\max_a Q_\pi(s,a) - J_\pi(s)|,.$$

Note that the Bellman error is the largest advantage, $\varepsilon = \max_{s,a}|A(s,a)|$. The Bellman error is usually defined with respect to value functions, but here we have generalized to a policy.

Theorem 4.3. *Let ε be the Bellman error of π and Δ be the l_∞ error of π. Then*

$$\Delta \leq \frac{\varepsilon}{1-\gamma}\,. \tag{4.5}$$

The proof rests on the following lemma which is useful equality.

Lemma 4.4. *For all (non-stationary) policies $\tilde{\pi}$, we have:*

$$\forall s\ J_{\tilde{\pi}}(s) - j_\pi(s) = \frac{1}{1-\gamma} E_{s,a\sim d(s|\tilde{\pi},\delta(s)),\tilde{\pi}}\{A_\pi(s,a)\}$$

Proof. Let d_t be the distribution of states implied by $\tilde{\pi}$ at time t.

For all s, we have:

$$\begin{aligned}
J_{\tilde{\pi}}(s) - J_\pi(s) &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t E_{s_t, a\sim d_t, \tilde{\pi}}\{\mathcal{R}(s_t,a)\} - J_\pi(s) \\
&= \sum_{t=0}^{\infty}\gamma^t E_{s_t, a\sim d_t, \tilde{\pi}}\{(1-\gamma)\mathcal{R}(s_t,a) + \gamma J_\pi(s_{t+1}) - J_\pi(s_t))\} \\
&= \sum_{t=0}^{\infty}\gamma^t E_{s_t, a\sim d_t, \tilde{\pi}}\{A(s_t,a)\} \\
&= \frac{1}{1-\gamma} E_{s,a\sim d(s|\tilde{\pi},\delta(s)),\tilde{\pi}}\{A_\pi(s,a)\}
\end{aligned}$$

□

Now, we will prove the theorem holds

Proof. (of theorem 4.3) Consider the non-stationary policy $\tilde{\pi} = \{\pi_0, \pi_1, \ldots \pi_{t-1}, \pi\}$, where the policy executes actions chosen according π_i for time i before t then executes π thereafter. Using lemma 4.4 we have that for all s:

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

In particular, this holds for the s which maximises $|J_{\tilde{\pi}}(s) - J_\pi(s)|$. Using the definition of ε, we get that:

$$\Delta \leq \frac{\varepsilon}{1-\gamma}$$

□

This bound is tight and is the best one could hope for some Bellman error (see appendix ** for an MDP that saturates this bound).

This bound elucidates the problems with using η as performance measure. For a given Bellman error, say ε, there is potential for $\frac{\varepsilon}{1-\gamma}$ improvement. However, the policy advantage any π' cannot be significantly large if there is no weight given to the states with large advantages. In contrast, using a μ_e such that the weighting $d_e(s)$ is not vanishingly small everywhere will guarantee that if there is a significant Bellman error than there exists a set of policies with a significant policy advantages, and so a correspondingly small number of samples will be needed to determine if π' improves ω.

Let's not hide the γ dependence here. -John

Corollary 4.5. *Let $d_0 = \min_s d(s|\pi, \mu_e)$. For a policy with Bellman error ε, there exists a policy π' such that $O\left((\frac{d_0\varepsilon}{n_a R})^2\right)$ are needed to determine if π' improves the error metric ω.*

Proof. The result follows from the fact that the maximal policy advantage must be at least $d_0\varepsilon$. *and? This is incomplete. -John* □

Hence, using ω with a more uniform μ_e will, in principle, only get stuck close to optimal, where all advantages must be small. *aside: Eventually (maybe here) make a comment that with prior knowledge it is possible to find a policy with large policy advantage using μ_e.*

5. A Polynomial Time Algorithm

We now address question 3, "in a reasonable number of steps, how good is the performance of the policy found?". First, we need a method for obtaining some π' to improve the policy with at each step, and we will assume that we have access to a black box that can give us a 'reasonably good' π'. Note that both gradient methods and greedy dynamic programming methods need to make this assumption as well. For greedy dynamic programming, π' must be chosen with zero l_∞-error to guarantee improvement, a far more stringent condition. Conversely, for gradient methods, the choice of parameterization implies the direction $\frac{\partial \pi}{\partial \theta}$ in policy space (see equation 4.2). There are few theoretical results for gradient methods which state conditions on the policy parameterization leading to the optimal policy.

We now explicitly state the conditions that our black box learning algorithm must satisfy. An ε good black box will always give us a policy with policy advantage ε close to optimal. More formally,

Definition 5.1. An ε black box algorithm is a function $BlackBox_\varepsilon(\pi, \mu)$ such that for all policies π and all start-state distributions μ, the function returns a π' such that $\Pi_{\pi,\mu}(\pi') \geq \max_{\tilde{\pi}} \Pi_{\pi,\mu}(\tilde{\pi}) - \varepsilon$.

This black box could fail to return a policy with positive policy advantage when the maximal policy advantage of the current policy is ε. If the maximum policy advantage of a policy π is $\varepsilon + \delta$, then we will be guaranteed a new policy π' with a policy advantage of at least δ. Note that the blackbox output can be easily verified. For a policy advantage of δ, we can verify that the policy advantage is positive with $O((\frac{n_a R}{\delta})^2)$ samples.

We now show that the number of calls to this black box needed to a obtain a policy with policy advantage less than or equal $\varepsilon + \delta$ is polynomial in $1/\delta$ and R. In section 5.1, we analyze how far we we can improve ω given a policy π'. If π' has a policy advantage of $\Pi > 0$, we can always obtain an improvement of at least $\frac{\Pi^2}{8R}$. Using this result we later give a polynomial bound on the number of calls to the black box required in order to reach a policy π for which no π' with a large policy advantage exists. Given this condition, we bound the quality of the policy found.

5.1. How much improvement do we get?

Given some policy π' with a policy advantage of $\Pi > 0$, how much can ω improve if we do a line search over α? We know that for sufficiently small α, there will be some policy improvement. At the other extreme of $\alpha = 1$, we know that there could significant degradation.

Theorem 5.2. *Let Π be the policy advantage of π' with respect to the policy π and the distribution μ_e and let ε be the Bellman error of π. For the update rule*

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

the change in the performance measure satisfies the following bound for all $\alpha \in [0, 1]$:

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

Before we state the proof, let us examine this result. The first term is analogous to the first order increase specified in theorem 4.1, and the second term is a penalty term based on the worst case scenario. 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

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

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 a deviant action from π', so if the current state distribution is $d(s|\pi, \mu_e)$ when the deviant action is chosen then 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 change in the performance can be written as

$$\omega(\pi_{\text{new}}) - \omega(\pi) = \frac{1}{1-\gamma} E_{a,s\sim\pi_{\text{new}}(a|s),d(s|\pi_{\text{new}},\mu_e,\gamma)} A_\pi(s,a)\,,$$

which follows from lemma 4.4 and linearity of expectation.

For the remainder of the proof we do not explicitly state the μ_e dependence.

Note that the term:

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

where we have used $\sum_a \pi(a|s) A_\pi(s,a) = 0$.

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'. 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 $\Pr(c_t \geq 1) = 1 - (1-\alpha)^t$. It follows that for all timesteps t:

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

where the inequalities follow from the definition of ε and $d_t(s|\pi_{\text{new}})$ is the distribution of states at time t given the start state distribution and a policy of π_{new}.

By substitution, we have:

$$\begin{aligned}
\omega(\pi_{\text{new}}) - \omega(\pi) &= \frac{1}{1-\gamma} E_{a,s\sim\pi_{\text{new}}(a|s),d(s|\pi_{\text{new}},\mu_e)} A_\pi(s,a) \\
&= \sum_{t=0}^{\infty} \gamma^t E_{a,s_t\sim\pi_{\text{new}}(a|s_t),d_t(s_t|\pi_{\text{new}},\mu_e)} \{A_\pi(s_t,a)\} \\
&\geq \sum_{t=0}^{\infty} \gamma^t (\alpha \sum_{s_t} d_t(s_t|\pi_{\text{new}},\mu_e,c_t=0) \sum_{a_t} \pi'(a_t|s_t) A_\pi(s_t,a_t) - \alpha 2(1-(1-\alpha)^t)\varepsilon) \\
&= \sum_{t=0}^{\infty} \gamma^t (\alpha \sum_{s_t} d_t(s_t|\pi,\mu_e) \sum_{a_t} \pi'(a_t|s_t) A_\pi(s_t,a_t) - \alpha 2(1-(1-\alpha)^t)\varepsilon) \\
&= \frac{\alpha}{(1-\gamma)} (E_{a,s\sim\pi'(a|s),d(s|\pi,\mu_e)} A_\pi(s,a) - 2\varepsilon \sum_{t=0}^{\infty} \gamma^t (1-(1-\alpha)^t) \\
&= \alpha \frac{\Pi_\pi(\pi')}{(1-\gamma)} - 2\alpha\varepsilon(\frac{1}{1-\gamma} - \frac{1}{1-\gamma(1-\alpha)})
\end{aligned}$$

where we have used the definitions of $d(s|\pi)$ and π. The result follows from simple algebra. □

Appendix 9 gives an example showing this bound is tight for all α. *John, this example is worth working through. Also, we need to say that things look bad for gradient methods, since there isn't a guarantee that for large improvement in those nasty cases where the policy advantage is small.*

Now let us analyse how far we can go. The only parameter in the previous bound that isn't necessarily known is ε_-, but this is bounded by R.

Corollary 5.3. *Under the assumptions in the previous equation, then*

$$\omega(\pi_{new}) - \omega(\pi) \geq \frac{\alpha}{1-\gamma}(\Pi - \alpha\frac{2R}{(1-\gamma)}) .$$

Further, if we assume that $\Pi \geq 0$, then this bound is maximized when $\alpha = \frac{(1-\gamma)\Pi}{4R}$ with a corresponding performance change of

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

Proof. The bound is a straightforward result of the previous theorem. Maximizing this lower bound we find $\alpha = \frac{(1-\gamma)\Pi}{4R}$, and substitution leads to the result. □

For large γ, the improvement of $\frac{\Pi^2}{8R}$ is close to the exact maximum of the tight bound (assuming $\varepsilon = R$), but for small γ, more improvement is possible in this worst case scenario.

We can now state the main theorem of the paper. Let us first define the maximal policy advantage with respect to π to be:

$$\Pi_{\text{opt},\mu}(\pi) = \max_{\pi'} \Pi_{\pi,\mu}(\pi')$$

where again we suppress the μ dependence if this is clear from context.

Theorem 5.4. *In at most $\frac{8R^2}{\delta^2}$ calls to an ε-black box algorithm, we can find a policy π' such that $\Pi_{opt}(\pi') < \varepsilon + \delta$.*

Proof. The worst case situation is if initially $\omega = 0$. If π' is not found yet, then the largest policy advantage must be greater than $\varepsilon + \delta$, so the black box must return a policy with $\Pi > \delta$. Hence, ω must improve by $\frac{\delta^2}{8R}$. Since ω can be at most R, the maximal number of calls is $\frac{R}{\Delta\omega} \le \frac{8R^2}{\delta^2}$. □

aside: I have left out the fact that we must check the black box to ensure that $\Pi > \delta$ at each step. If not, the Black Box is close to broken, and are done, ie we found a policy such that $\Pi_{\text{opt}}(\pi') < \varepsilon + \delta$. This verification is also easy.

5.2. **How good is the policy we find?**

Theorem 5.5. *Assume we have some π' such that $\Pi_{opt}(\pi') < \varepsilon + \delta$ and let $d_0 = \min_s d(s|\pi', \mu_e)$. Then*

$$\max_s |J_{\pi'}(s) - J_{opt}(s)| \le \frac{\varepsilon + \delta}{d_0(1-\gamma)} .$$

Proof. Let b be the Bellman error and let s' be the corresponding state. Then $\Pi_{\text{opt}}(\pi') = \sum_s d(s) \max_a A_{\pi'}(s,a) \ge d(s')b \ge d_0 b$. Therefore, the Bellman error is upper bounded by $\frac{\varepsilon+\delta}{d_0}$, and the result follows from theorem 4.3. □

Note that our exploratory distribution is necessary to keep d_0 from becoming unreasonably small.

aside: We need some continuity conditions to make the d_0 factor look less nasty! OTOH, the factor of d_0 is much more nasty for gradient methods since this could unboundedly small.

6. Connections to η and continuity

We need to give a few results here about the connection to η, like how the choice

$$\mu_{\text{mix}} = \mu_e + \mu ,$$

is a good choice.

We should also state something about continuity conditions.

7. Discussion

(1) discuss sensible supervised learning methods.
(2) how to represent compact policies?
(3) discuss why the assumption that we have access to an exploration policy is reasonable.

8. other stuff

(1) Note that this update rule is to first order equal to the update rule $\pi(a|s) \propto \pi(a|s) \exp\left(\alpha \frac{f(s,a)}{\pi(s,a)}\right)$, which enforces positivity.

References

9. Appendix

The goal of this section is constructing a proof that policy gradient will *eventually* work on a Gibbs table lookup distribution. Note that the time scale of "eventually" can be remarkably long.

The Gibbs table-lookup distribution is $\tilde{\Pi}_G = \{\pi_\theta : \pi(a|s) \propto \exp(\theta_{sa})\}$. Using the notation $\partial_{sa} \equiv \partial / \partial\theta_{sa}$,

$$\partial_{sa}\eta(\pi) = d_\pi(s)\pi(a|s)(Q_\pi(s,a) - J_\pi(s)) = d_\pi(s)\pi(a|s)A_\pi(s,a) \tag{9.1}$$

Note that for any deterministic policy (*ie* $\forall s\, \exists a$ s.t. $\theta_{sa} = \infty$) the gradient is zero. Since the gradient is 0 at all points at infinity, this suggests the dismal possibility that the gradient descent could lead to arbitrary deterministic policy. It is straight forward to show that if the stationary distribution, of the asymptotic policy, is nonzero for all states then this asymptotic policy is optimal. However, the non-trivial question is can $\partial_{sa}\eta_\pi$ tend to 0 due to $d(s)$ tending to 0 in a manner such that such that the optimal policy is not achieved. Note that the gradient $\partial_{sa}\eta(\pi)$ is zero if either the advantage is zero or if $d(s)$ is zero. The following theorem states that this can never occur, and the key to proof is dealing with this case (where learning to one state could stop due to a vanishing stationary probability). We only consider the infinitesimal learning rate case where $\theta_t = \theta_0 + \int_0^t \nabla\eta^{\pi_t}\, dt$.

Theorem 9.1. *Assume that π_∞ is the single limiting policy corresponding to $\lim_{t\to\infty}\theta_t$. Then π_∞ is globally optimal.*

Proof. Let $S^* = \{s | \exists \pi \text{ s.t. } \rho_\pi(s) \neq 0\}$. Our proof that π_∞ globally maximizes the average reward consists of showing that for all $s \in S^*$, $J^{\pi_\infty}(s) = \max_a Q^{\pi_\infty}(s,a)$, *ie* π_∞ partially satisfies the Bellman equations. When clear, the subscript on π_t is dropped, *eg* $\rho^{\pi_\infty} = \rho^\infty$. Assume for now that as $t \to \infty$ then for all $s \in S^*$ and actions a either $\pi_t(a|s) \to 0$ or $A^t(s,a) \to 0$. For all $\tilde{a} \in \operatorname{argmax}_a Q^{\pi_\infty}(s,a)$, $\partial_{sa}\eta^t$ is positive for sufficiently large t, so $\pi_\infty(a|s) \neq 0$ and $A^\infty(s,a) = 0$. Therefore, π_∞ satisfies the Bellman equations for $s \in S^*$. It follows that if $\tilde{a} \notin \operatorname{argmax}_a Q^{\pi_\infty}(s,a)$ then $\pi_\infty(\tilde{a}, s) = 0$, and so if t is sufficiently large $\partial_{s\tilde{a}}\eta^t < 0$, else there must exist a partial extrema for some $\tilde{t} \geq t$.

It is straightforward to show that $\lim_{t\to\infty}\partial_{sa}\eta^t = 0$, and so as $t \to \infty$ either $\rho^t \to 0$, $\pi_t(a|s) \to 0$, or $A^t(s,a) \to 0$. We now show that even if if $\rho^\infty \neq 0$, then $\pi_\infty(a|s) = 0$ or $A^\infty(s,a) = 0$, which completes the proof. Assume that for some s that $\rho^\infty(s) = 0$. By stationarity, $\rho^t(s) = \sum_{s_0} P(s_\tau = s|s_0)\rho^t(s_0)$, and so for any trajectory $s_0, a_0, s_1, a_1 \dots s$, $\rho^t(s) \geq \rho^t(s_0)p^{a_0}_{s_0 s_1}\pi_t(a_0|s_0)\dots p^{a_{\tau-1}}_{s_{\tau-1}s}\pi_t(a_{\tau-1}|s_{\tau-1})$. For every τ-length trajectory with $p^{a_i}_{s_i s_{i+1}} \neq 0$ and $\rho^\infty(s_0) \neq 0$, there must exist at least one s_i, a_i such that $\pi_\infty(a_i|s_i) = 0$, else $\rho^\infty(s) \neq 0$ or $s \notin S^*$. It can be shown that there exist some $\tilde{s}$, $\tilde{a}$ and positive constant $\tilde{\alpha}$ such that $\forall t\, \rho^t(s) \geq \tilde{\alpha}\rho^t(\tilde{s})\pi_t(\tilde{a}|\tilde{s})$, $\rho^\infty(\tilde{s}) \neq 0$, and $\pi_\infty(\tilde{a}|\tilde{s}) = 0$ (using a trajectory such that $\pi_\infty(a_i|s_i) = 0$ only for $i = 0$). Assume that $\pi_\infty(s) \neq 0$ and $A^\infty(s) \neq 0$ (else the conjecture is true), then it follows from equation 9.1 and the previous inequality that for each a, $|\int_0^\infty \partial_{sa}\eta^t dt| \geq \alpha |\int_0^\infty \partial_{\tilde{s}\tilde{a}}\eta^t dt|$ for some positive constant α. However, $\pi_\infty(\tilde{s}) = 0$ forces $|\int_0^\infty \partial_{\tilde{s}\tilde{a}}\eta^t dt|$ to be ∞, and so $|\int_0^\infty \partial_{sa}\eta^t dt|$ is ∞, contradicting $\pi_\infty(s) \neq 0$ and $A^\infty(s) \neq 0$. $\square$

10. Appendix B: Examples

The example for theorem 4.3 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 $A(i,1) = 0$, $A(i,2) = R$, $A(j,1) = 0$, and $A(j,2) = -R$. Let us choose π' such that $\pi'(2|i) = 1$ and $\pi'(2|j) = 1$, so:

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

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

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

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

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

11. Policy Improvement

There is no reason to believe that a "good" choice of the vector $f(s,a)$ can be made without significant prior knowledge about the form of the reinforcement learning problem. Given some number of samples drawn from $\nu(s,a)$, we can test the value of a particular vector $f(s,a)$. The vector f or $-f$ may satisfy: $E_\nu fA > \epsilon$, in which case we can improve the policy with high probability. Or, it may be that the expectation is to small and we will be unable to determine whether or not the f is a beneficial perturbation given the (necessarily finite) samples from $\nu(s,a)$. This last case is by far the most common, and so it is necessary that we test many different f with the same set of samples from $\nu(s,a)$. This necessity creates the possibility of overfitting: we might find an f which works *only* on our sample set and not on future draws from $\nu(s,a)$.

There are two approaches to avoiding overfitting. One is to work with a finitely parameterized set F of vectors $f(s,a)$, and then simply use enough samples from $\nu(s,a)$ that we can guarantee (via VC bounds or other approaches) that the empirical estimate of $E_\nu fA$ must be strongly related to the actual value of $E_\nu fA$. The disadvantage of this approach is that it might not work. In particular, it is unclear (without extra prior knowledge) that we can guarantee there exists $f \in F$ with a large dot product ($E_\nu fA$). This is exactly what we wish to avoid by shifting from optimizing η to ω.

The other approach is "non-parametric" which is a great misnomer. In a non-parametric approach the number of parameters grows with the number of samples, but always remains small enough that we can guarantee that the empirical estimate of $E_\nu fA$ remains close to its real value. This is the approach that we advocate here: choose an $f(s,a)$ by sampling from $\nu(s,a)$ and testing until an $f(s,a)$ is found which improves ω with high probability. In the process of discovering $f(s,a)$, increase the space of functions as more samples are acquired. We can *always* check to see if a "good" $f(s,a)$ has been learned by simply acquiring a few extra independent samples from $\nu(s,a)$ and testing to see that $f(s,a)$ behaves well on the extra samples. If the test fails, then we have more samples from which to learn a new f.

Our suggested technique then does the following:

(1) Sample k samples from $\nu(s,a)$ and calculate $\hat{Q}(s,a)$ for each sample.
(2) Find an $f(s,a) \in F(k)$ for which the empirical estimate of $\frac{1}{k}\sum_{i=1}^{k} \bar{f}(s,a)Q(s,a)$ is largest.
(3) Verify that $f(s,a)$ does well with l new independent samples drawn from $\nu(s,a)$ (or other sample complexity approaches if applicable). If no, then go to step 2 with an extra l samples.
(4) Alter the policy by $\pi(a|s) \propto \pi(a|s) + \alpha f(s,a)$ and go to step 1.

This approach will always result in improvement according to the metric ω and so it will always result in improvement given that significant advantages exist. The important point is that the verification step can be done efficiently for an ε-far policy, as stated by **??**.

We need to connect this general framework more tightly to the previous results. -Sham

Another advantage of the non-parametric approach is that we can avoid trading off between improvement of ω and improvement of η. Samples from η are easier to acquire so we could easily acquire an addition k samples from η and then find the optimal f for the empirical distribution which is a mix of the $\nu(s,a)$ and $d_\pi(s)\pi(a|s)$. If the choice of f makes optimization on the $d_\pi(s)\pi(a|s)$ samples essentially orthogonal to optimization on the $\nu(s,a)$ samples, then we expect that $f(s,a)$ will almost never decrease the value η.

12. How to Monte Carlo

In order to determine which direction is the "right" direction, we need to know how to estimate this right direction. This is a subtle problem when you must (in general) sample from the underlying reinforced MDP forever in order to acquire an unbiased estimate of the discounted future reward, $\hat{Q}(s,a)$.

Intuitively, if we run the Markov process for a time long compared to the discount γ, we will obtain an estimate $\hat{Q}(s,a)$ which is "nearly" unbiased, simply due to the fact that the contribution to the reward from future actions will necessarily be small.