CS234 Notes - Lecture 9
Advanced Policy Gradient

Patrick Cho, Emma Brunskill

February 11, 2019

1 Policy Gradient Objective

Recall that in Policy Gradient, we parameterize the policy π_θ and directly optimize for it using experience in the environment. We first define the probability of a trajectory given our current policy π_θ, which we denote as $\pi_\theta(\tau)$.

$$\pi_\theta(\tau) = \pi_\theta(s_1, a_1, ..., s_T, a_T) = P(s_1)\prod_{t=1}^{T}\pi_\theta(a_t|s_t)P(s_{t+1}|s_t, a_t)$$

Parsing the function above, $P(s_1)$ is the probability of starting at state s_1, $\pi_\theta(a_t|s_t)$ is the probability of our current policy selecting action a_t given that we are in state s_t, and $P(s_{t+1}|s_t, a_t)$ is the probability of the environment's dynamics transiting us to state s_{t+1} given that we start at s_t and take action a_t. Note that the we overload the notation for π_θ here to either mean the probability of a trajectory ($\pi_\theta(\tau)$) or the probability of an action given a state ($\pi_\theta(a|s)$).

The goal of Policy Gradient, similar to most other RL objectives that we have discussed thus far, is to maximize the discounted sum of rewards.

$$\theta^* = \arg\max_\theta \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \gamma^t r(s_t, a_t)\right]$$

We denote our objective function as $J(\theta)$ which can be estimated using Monte Carlo. We also use $r(\tau)$ to represent the discounted sum of rewards over trajectory τ.

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \gamma^t r(s_t, a_t)\right] = \int \pi_\theta(\tau)r(\tau)d\tau \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\gamma^t r(s_{i,t}, a_{i,t})$$

$$\theta^* = \arg\max_\theta J(\theta)$$

We define $P_\theta(s, a)$ to be the probability of seeing (s,a) pair in our trajectory. Note that in the case of infinite horizon where a stationary distribution of states exist, we can write $P_\theta(s, a) = d^{\pi_\theta}(s)\pi_\theta(a|s)$ where $d^{\pi_\theta}(s)$ is the stationary state distribution under policy π_θ.

In the infinite horizon case, we have

$$\begin{aligned}\theta^* &= \arg\max_\theta \sum_{t=1}^{\infty}\mathbb{E}_{(s,a)\sim P_\theta(s,a)}[\gamma^t r(s, a)] \\ &= \arg\max_\theta \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim P_\theta(s,a)}[r(s, a)] \\ &= \arg\max_\theta \mathbb{E}_{(s,a)\sim P_\theta(s,a)}[r(s, a)]\end{aligned}$$

In the finite horizon case, we have

$$\theta^* = \arg\max_\theta \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim P_\theta(s_t,a_t)}[\gamma^t r(s_t, a_t)]$$

We can use gradient based methods to do the above optimization. In particular, we need to find the gradient of $J(\theta)$ with respect to θ.

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \int \pi_\theta(\tau) r(\tau) d\tau \\
&= \int \nabla_\theta \pi_\theta(\tau) r(\tau) d\tau \\
&= \int \pi_\theta(\tau) \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} r(\tau) d\tau \\
&= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) r(\tau)]
\end{aligned}$$

As seen above, we have moved the gradient from outside of the expectation to inside of the expectation. This is commonly known as the log derivative trick. The advantage of doing so is that now we do not need to take gradient over the dynamics function as seen below.

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) r(\tau)] \\
&= \mathbb{E}_{\tau\sim\pi_\theta(\tau)} \left[\nabla_\theta \left[\log P(s_1) + \sum_{t=1}^{T} (\log \pi_\theta(a_t|s_t) + \log P(s_{t+1}|s_t, a_t)) \right] r(\tau) \right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta(\tau)} \left[\nabla_\theta \left[\sum_{t=1}^{T} (\log \pi_\theta(a_t|s_t)) \right] r(\tau) \right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta(\tau)} \left[\sum_{t=1}^{T} \left(\nabla_\theta (\log \pi_\theta(a_t|s_t)) \left(\sum_{t=1}^{T} \gamma^t r(s_t, a_t) \right) \right) \right] \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\nabla_\theta (\log \pi_\theta(a_{i,t}|s_{i,t})) \left(\sum_{t=1}^{T} \gamma^t r(s_{i,t}, a_{i,t}) \right) \right)
\end{aligned}$$

In the third equality, the terms cancel out because they do not involve θ. In the last step, we use Monte Carlo estimates from rollout trajectories.

Note that there are many similarities between the above formulation and the Maximum Likelihood Estimate (MLE) in the supervised learning setting. For MLE in supervised learning, we have likelihood, $J'(\theta)$, and log-likelihood, $J(\theta)$:

$$J'(\theta) = \prod_{i=1}^{N} P(y_i|x_i)$$

$$J(\theta) = \log J'(\theta) = \sum_{i=1}^{N} \log P(y_i|x_i)$$

$$\nabla_\theta J(\theta) = \sum_{i=1}^{N} \nabla_\theta \log P(y_i|x_i)$$

Comparing with the Policy Gradient derivation, the key difference is the sum of rewards. We can even view MLE as policy gradient with a return of 1 for all examples. Although this difference may seem minor, it can cause the problem to become much harder. In particular, the summation of rewards drastically increases variance. Hence, in the next section, we discuss two methods to reduce variance.

2 Reducing Variance in Policy Gradient

2.1 Causality

We first note that the action taken at time t' cannot affect reward at time t for all $t < t'$. This is known as causality since what we do now should not affect the past. Hence, we can change the summation of rewards, $\sum_{t=1}^{T} \gamma^t r(s_{i,t}, a_{i,t})$, to the reward-to-go, $\hat{Q}_{i,t} = \sum_{t'=t}^{T} \gamma^{t'} r(s_{i,t'}, a_{i,t'})$. We use $\hat{Q}$ here to denote that this is a Monte Carlo estimate of Q. Doing so helps to reduce variance since we effectively reduce noise from prior rewards. In particular, our objective changes to:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\nabla_\theta \log \pi_\theta(a_{i,t}, s_{i,t}) \left(\sum_{t'=t}^{T} \gamma^{t'} r(s_{i,t'}, a_{i,t'}) \right) \right) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\nabla_\theta \log \pi_\theta(a_{i,t}, s_{i,t}) \hat{Q}_{i,t} \right)$$

2.2 Baselines

Now, we consider subtracting a baseline from the reward-to-go. That is, we change our objective into the following form:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_{i,t}, s_{i,t}) \left[\left(\sum_{t'=t}^{T} \gamma^{t'} r(s_{i,t'}, a_{i,t'}) \right) - b \right]$$

We first note that subtracting a constant baseline, b, is unbiased. That is under expectation of trajectories from our current policy π_θ, the term we have just included is 0.

$$\begin{aligned}
\mathbb{E}_{\tau \sim \pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) b] &= \int \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) b d\tau \\
&= \int \pi_\theta(\tau) \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} b d\tau \\
&= \int \nabla_\theta \pi_\theta(\tau) b d\tau \\
&= b \nabla_\theta \int \pi_\theta(\tau) d\tau \\
&= b \nabla_\theta 1 = 0
\end{aligned}$$

In the last equality, the integral of the probability of a trajectory over all trajectories is 1. In the second last equality, we are able to take b out of the integral since b is a constant (e.g. average return, $b = \frac{1}{N} \sum_{i=1}^{N} r(\tau)$). However, we can also show that this term is unbiased if b is a function of state s.

$$\begin{aligned}
\mathbb{E}_{\tau \sim \pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(a_t|s_t) b(s_t)] &= \mathbb{E}_{s_{0:t}, a_{0:(t-1)}} \left[\mathbb{E}_{s_{(t+1):T}, a_{t:(T-1)}} [\nabla_\theta \log \pi_\theta(a_t|s_t) b(s_t)] \right] \\
&= \mathbb{E}_{s_{0:t}, a_{0:(t-1)}} \left[b(s_t) \mathbb{E}_{s_{(t+1):T}, a_{t:(T-1)}} [\nabla_\theta \log \pi_\theta(a_t|s_t)] \right] \\
&= \mathbb{E}_{s_{0:t}, a_{0:(t-1)}} \left[b(s_t) \mathbb{E}_{a_t} [\nabla_\theta \log \pi_\theta(a_t|s_t)] \right] \\
&= \mathbb{E}_{s_{0:t}, a_{0:(t-1)}} \left[b(s_t) \cdot 0 \right] = 0
\end{aligned}$$

As seen above, if no assumptions on the policy are made, the baseline cannot be a function of actions since the proof depends on being able factor out $b(s_t)$. Exceptions exist if we make some assumptions. See [3] for an example of action-dependent baselines.

One common baseline that is used is the value function, $V^{\pi_\theta}(s)$. Since the reward-to-go estimates the state-action value function $Q^{\pi_\theta}(s,a)$, by subtracting this baseline from Q, we are essentially calculating the advantage, $A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$. In terms of implementation, this means training a separate value function $V_\phi(s)$.

As a side note, instead of using actual returns from the environment to estimate $Q^{\pi_\theta}(s,a)$, we can train another state-action value function $Q_w(s,a)$ to approximate the policy gradient. This approach is known as actor-critic where the Q_w function is the critic. Essentially, the critic does policy evaluation and the actor does policy improvement.

One can ask: what is the optimal baseline to subtract in order to minimize variance? The optimal baseline is in fact the expected reward weighted by the square of the gradients as shown below.

$$Var[x] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$
$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau)(r(\tau) - b)]$$

$$\begin{aligned} Var &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[(\nabla_\theta \log \pi_\theta(\tau)(r(\tau) - b))^2\right] - \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\nabla_\theta \log \pi_\theta(\tau)(r(\tau) - b)\right]^2 \\ &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[(\nabla_\theta \log \pi_\theta(\tau)(r(\tau) - b))^2\right] - \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\nabla_\theta \log \pi_\theta(\tau)(r(\tau))\right]^2 \end{aligned}$$

In the equation above, we are able to remove b in the second term since we have proven that b is unbiased in expectation. To minimize variance, we set its gradient with respect to b to 0. The second term in the variance equation does not depend on b and therefore disappears.

$$\begin{aligned} \frac{dVar}{db} &= \frac{d}{db}\mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[(\nabla_\theta \log \pi_\theta(\tau)(r(\tau) - b))^2\right] \\ &= \frac{d}{db}(-2\mathbb{E}_{\tau\sim\pi_\theta(\tau)}[(\nabla_\theta \log \pi_\theta(\tau))^2 r(\tau) b] + \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[(\nabla_\theta \log \pi_\theta(\tau))^2 b^2]) \\ &= -2\mathbb{E}[(\nabla_\theta \log \pi_\theta(\tau))^2 r(\tau)] + 2\mathbb{E}[(\nabla_\theta \log \pi_\theta(\tau))^2 b] = 0 \end{aligned}$$

$$b = \frac{\mathbb{E}[(\nabla_\theta \log \pi_\theta(\tau))^2 r(\tau)]}{\mathbb{E}[(\nabla_\theta \log \pi_\theta(\tau))^2]}$$

3 Off Policy Policy Gradient

In the analysis above, our objective involves taking an expectation over trajectories drawn from $\pi_\theta(\tau)$. This means that Policy Gradient with the above objective will result in an on policy algorithm. Whenever we change our parameters θ, our policy changes and all our old trajectories cannot be reused. Compare this with DQN which is able to store prior experience to be reused since it is an off policy algorithm. Hence, Policy Gradient is in general less sample efficient than Q learning. To resolve this, we discuss the use of Importance Sampling to generate an Off Policy Policy Gradient algorithm. In particular, we consider the changes that need to be made if we estimate $J(\theta)$ using trajectories drawn from a prior policy π_θ instead of current policy $\pi_{\theta'}$.

$$
\begin{aligned}
\theta^* &= \arg\max_{\theta'} J(\theta') \\
&= \arg\max_{\theta'} \mathbb{E}_{\tau\sim\pi_{\theta'}(\tau)}[r(\tau)] \\
&= \arg\max_{\theta'} \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}\left[\frac{\pi_{\theta'}(\tau)}{\pi_{\theta}(\tau)} r(\tau)\right] \\
&= \arg\max_{\theta'} \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}\left[\frac{P(s_1)\prod_{t=1}^{T}\pi_{\theta'}(a_t|s_t)P(s_{t+1}|s_t,a_t)}{P(s_1)\prod_{t=1}^{T}\pi_{\theta}(a_t|s_t)P(s_{t+1}|s_t,a_t)} r(\tau)\right] \\
&= \arg\max_{\theta'} \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}\left[\frac{\prod_{t=1}^{T}\pi_{\theta'}(a_t|s_t)}{\prod_{t=1}^{T}\pi_{\theta}(a_t|s_t)} r(\tau)\right]
\end{aligned}
$$

Hence, we have for old parameters θ:

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}[r(\tau)]$$

and for new parameters θ':

$$J(\theta') = \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}[\frac{\pi_{\theta'}(\tau)}{\pi_{\theta}(\tau)} r(\tau)]$$

$$
\begin{aligned}
\nabla_{\theta'} J(\theta') &= \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}\left[\frac{\nabla_{\theta'}\pi_{\theta'}(\tau)}{\pi_{\theta}(\tau)} r(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}\left[\frac{\pi_{\theta'}(\tau)}{\pi_{\theta}(\tau)} \nabla_{\theta'}\log\pi_{\theta'}(\tau) r(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}\left[\left(\prod_{t=1}^{T}\frac{\pi_{\theta'}(a_t|s_t)}{\pi_{\theta}(a_t|s_t)}\right)\left(\sum_{t=1}^{T}\nabla_{\theta'}\left(\log\pi_{\theta'}(a_t|s_t)\right)\right)\left(\sum_{t=1}^{T}\gamma^t r(s_t,a_t)\right)\right] \\
&= \mathbb{E}_{\tau\sim\pi_{\theta}(\tau)}\left[\sum_{t=1}^{T}\left(\nabla_{\theta'}\left(\log\pi_{\theta'}(a_t|s_t)\right)\left(\prod_{t'=1}^{t}\frac{\pi_{\theta'}(a_{t'}|s_{t'})}{\pi_{\theta}(a_{t'}|s_{t'})}\right)\left(\sum_{t'=t}^{T}\gamma^{t'} r(s_{t'},a_{t'})\right)\right)\right]
\end{aligned}
$$

In the last equality, we invoke causality. In particular, the probability of the first k transitions only depends on the first k actions and not future actions.

4 Relative Policy Performance Identity

One issue with directly taking gradient steps according to the gradient of the objective $J(\theta)$ with respect to the parameters θ is that moving in the parameter space is not the same as moving in the policy space. This causes a problem in the choice of step sizes. Small step sizes causes learning to be slow but large step sizes may cause the policy to become bad.

In the case of supervised learning, this is usually fine since the following updates will usually fix this problem. However, in the context of reinforcement learning, a bad policy will cause the next batch of data to be collected under the bad policy. Hence, stepping to a bad policy may cause a collapse in performance from which the algorithm cannot recover. Simple line search in the direction of the gradient could be performed to mitigate this issue. For example, we could try multiple learning rates

for each update and choose the learning rate that gives best performance. However, doing so is naive and will cause slow convergence in cases where the first order approximation (gradient) is bad.

Trust Region Policy Optimization, discussed in the next section, is an algorithm that tries to resolve this issue. Building towards this, we first derive an identity with regards to relative policy performance, that is $J(\pi') - J(\pi)$. Here we use the following notations: $J(\pi') = J(\theta')$, $J(\pi) = J(\theta)$, $\pi' = \pi_{\theta'}$ and $\pi = \pi_\theta$.

Lemma 4.1.

$$J(\pi') - J(\pi) = \mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^\pi(s_t, a_t)\right]$$

Proof.

$$\begin{aligned}
\mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^\pi(s_t, a_t)\right] &= \mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t [r(s_t, a_t) + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)]\right] \\
&= \mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t [r(s_t, a_t)]\right] + \mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t [\gamma V^\pi(s_{t+1}) - V^\pi(s_t)]\right] \\
&= J(\pi') - \mathbb{E}_{\tau\sim\pi'}[V^\pi(s_0)] \\
&= J(\pi') - J(\pi)
\end{aligned}$$

□

Hence, we have

$$\begin{aligned}
\max_{\pi'} J(\pi') &= \max_{\pi'} J(\pi') - J(\pi) \\
&= \max_{\pi'} \mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^\pi(s_t, a_t)\right]
\end{aligned}$$

The issue with the above expression is that we require trajectories from π'. This makes optimization impossible since we have not yet found π' but need to draw samples from π'. Once again, we use likelihood ratios to circumvent this issue.

$$\begin{aligned}
J(\pi') - J(\pi) &= \mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^\pi(s_t, a_t)\right] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi'} \\ a\sim\pi'}}[A^\pi(s, a)] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi'} \\ a\sim\pi}}[\frac{\pi'(a|s)}{\pi(a|s)}A^\pi(s, a)] \\
&\approx \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi} \\ a\sim\pi}}[\frac{\pi'(a|s)}{\pi(a|s)}A^\pi(s, a)] \\
&= \frac{1}{1-\gamma}L_\pi(\pi')
\end{aligned}$$

We call $L_\pi(\pi')$ the surrogate objective. One key question is when we can make the above approximation. Clearly, when $\pi = \pi'$, the approximation holds with equality. However, this would not be useful since we want to improve our current policy π to a better policy π'. In the derivations of Trust Region Policy Optimization (TRPO) below, we give bounds for the approximation.

5 Trust Region Policy Optimization

The key idea in TRPO [5] is to define a trust region that constrains updates to the policy. This constraint is in the policy space rather than in the parameter space and becomes the new "step size" of the algorithm. In this way, we can approximately ensure that the new policy after the policy update performs better than the old policy.

5.1 Problem Setup

Consider a finite state and action MDP $\mathcal{M} = (S, A, M, R, \gamma)$ where M is the transition function. In this section, we assume that $|S|$ and $|A|$ are both finite, and that $0 < \gamma < 1$. Although the derivation is for finite states and actions, the algorithm works for continuous states and actions as well. We define

$$d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t M(s_t = s|\pi) \tag{1}$$

to be the discounted state visitation distribution following policy π with dynamics M starting at state s, and

$$V^\pi = \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^\pi \\ a\sim\pi(\cdot|s) \\ s'\sim M(\cdot|s,a)}}[R(s,a,s')] \tag{2}$$

to be the discounted expected sum of rewards when following policy π on with transition dynamics M. Note that V^π here has the same definition as $J(\theta)$ from the previous sections.

Let $\rho_\pi^t \in \mathbb{R}^{|S|}$ where $\rho_\pi^t(s) = M(s_t = s|\pi)$. This is the probability of being in state s at timestep t when following policy π with dynamics M.

Let $P_\pi \in \mathbb{R}^{|S|\times|S|}$ where $P_\pi(s'|s) = \sum_a M(s'|s,a)\pi(a|s)$. This is the probability of transitioning from state s to next state s' in one step by taking actions following policy π and using transitions from dynamics M.

Let μ be starting state distribution for $\mathcal{M}$. Then, we have:

$$\begin{aligned} d^\pi &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t M(s_t = s|\pi) \\ &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P_\pi\mu \\ &= (1-\gamma)(I-\gamma P_\pi)^{-1}\mu \end{aligned} \tag{3}$$

where the second equality holds true because $\rho_\pi^t = P_\pi\rho_\pi^{t-1}$ and the third equality can be derived from geometric series.

The goal in our proof is to give a lower bound for $V^{\pi'} - V^\pi$. We start our proof with a lemma on reward shaping.

Lemma 5.1. For any function $f : S \mapsto \mathbb{R}$ and any policy π, we have:

$$(1-\gamma)\mathop{\mathbb{E}}_{s\sim\mu}[f(s)] + \mathop{\mathbb{E}}_{\substack{s\sim d^\pi \\ a\sim\pi(\cdot|s) \\ s'\sim M(\cdot|s,a)}}[\gamma f(s')] - \mathop{\mathbb{E}}_{s\sim d^\pi}[f(s)] = 0 \tag{4}$$

The proof can be found in [4] and is reproduced in Section A.1 of the Appendix.

We can add this term to the R.H.S. of equation (2). Doing so, we get

$$V^{\pi}(s) = \frac{1}{1-\gamma}\left(\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)\\ s'\sim M(\cdot|s,a)}}[R(s,a,s') + \gamma f(s') - f(s)]\right) + \mathop{\mathbb{E}}_{s\sim\mu}[f(s)] \tag{5}$$

This can be seen as a form of reward shaping where the shaping function is only a function of states and not actions. Notice that if we substitute $f(s) = V^{\pi}(s)$, we get the advantage function.

5.2 Bounding Difference in State Distributions

When we update $\pi \to \pi'$, we will have different discounted state visitation distributions, d^{π} and $d^{\pi'}$, respectively. Now let us bound their difference.

Lemma 5.2.

$$\left\|d^{\pi'} - d^{\pi}\right\|_1 \leq \frac{2\gamma}{1-\gamma}\left[\mathop{\mathbb{E}}_{s\sim d^{\pi}}[D_{TV}(\pi'\|\pi)[s]]\right]$$

The proof can be found in [4] and is reproduced in Section A.2 of the Appendix.

5.3 Bounding Difference in Returns

Now, we bound the difference in value for the update $\pi \to \pi'$, namely

$$V^{\pi'} - V^{\pi}$$

Lemma 5.3. Defining the following terms,

$$L_{\pi}(\pi') = \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)}}\left[\frac{\pi'(a|s)}{\pi(a|s)}A^{\pi}(s,a)\right]$$

$$\epsilon_f^{\pi'} = \max_s\left[\mathop{\mathbb{E}}_{\substack{a\sim\pi'(\cdot|s)\\ s'\sim M(\cdot|s,a)}}[R(s,a,s') + \gamma f(s') - f(s)]\right]$$

we get the following upper bound

$$V^{\pi'} - V^{\pi} \leq \frac{1}{1-\gamma}\left(L_{\pi}(\pi') + \left\|d^{\pi'} - d^{\pi}\right\|_1 \epsilon_f^{\pi'}\right) \tag{6}$$

and the following lower bound

$$V^{\pi'} - V^{\pi} \geq \frac{1}{1-\gamma}\left(L_{\pi}(\pi') - \left\|d^{\pi'} - d^{\pi}\right\|_1 \epsilon_f^{\pi'}\right) \tag{7}$$

The proof can be found in [4] and is reproduced in Section A.3 of the Appendix.

We remind the reader that an upper bound for $\left\|d^{\pi'} - d^{\pi}\right\|_1$ is given in Lemma (5.2) which can be substituted into (6) and (7).

5.4 Bounding Maximum Advantage

Now we need to consider the term

$$\epsilon_f^{\pi'} = \max_s \left| \mathop{\mathbb{E}}_{\substack{a\sim\pi'(\cdot|s)\\ s'\sim M(\cdot|s,a)}} [R(s,a,s') + \gamma f(s') - f(s)] \right| \tag{8}$$

We set $f(s) = V^\pi(s)$ to be the value function at state s following policy π. Hence, we have

$$\epsilon_{V^\pi}^{\pi'} = \max_s \left| \mathop{\mathbb{E}}_{a\sim\pi'(\cdot|s)} [A^\pi(s,a)] \right| \tag{9}$$

This allows us to formulate the following:

Lemma 5.4.

$$\epsilon_{V^\pi}^{\pi'} \leq 2 \max_s D_{TV}(\pi \| \pi') \max_{s,a} |A^\pi(s,a)|$$

The proof can be found in [5] and is reproduced in Section A.4 of the Appendix.

5.5 TRPO

To recap, setting $f = V^\pi$, we have

$$\frac{1}{1-\gamma} L_\pi(\pi') - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2 \leq V^{\pi'} - V^\pi \leq \frac{1}{1-\gamma} L_\pi(\pi') + \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2 \tag{10}$$

where

$$L_\pi(\pi') = \mathop{\mathbb{E}}_{\substack{s\sim d^\pi\\ a\sim\pi(\cdot|s)}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^\pi(s,a) \right]$$

$$\epsilon = \max_{s,a} |A^\pi(s,a)|$$

$$\alpha = \max_s D_{TV}(\pi \| \pi')$$

Comparing the above equation with the equation we got in the previous section on Relative Policy Performance Identity, we have given lower bounds and upper bounds rather than just an approximation. By optimizing the lower bound of $V^{\pi'} - V^\pi$, we get an optimization problem that guarantees improvement to our policy. Concretely, we solve the following optimization problem:

$$\max_{\pi'} L_\pi(\pi') - \frac{4\epsilon\gamma}{(1-\gamma)}\alpha^2$$

Unfortunately, solving this optimization problem results in very small step sizes. In [5], the authors change this optimization problem to a constraint optimization problem to get larger step sizes when implementing a practical algorithm. Concretely, this results in the following optimization problem:

$$\max_{\pi'} L_\pi(\pi') \quad \text{s.t. } \alpha^2 \leq \delta$$

where δ is a hyperparameter.

The max constraint in α is impractical to solve due to the large number of states. Hence, in [5], the authors use a heuristic approximation which considers the average KL divergence only. This approximation is useful since we can approximate expectation with samples but we cannot approximate max with samples. Hence, we have

$$\max_{\pi'} L_\pi(\pi')$$
$$\text{s.t. } \bar{D}_{KL}(\pi, \pi') \leq \delta$$

where $\bar{D}_{KL}(\pi, \pi') = \mathbb{E}_{s\sim d^\pi}[D_{KL}(\pi \| \pi')[s]]$

6 Exercises

Exercise 6.1. In the lecture slides, for episodic environments, the objective function is given as $J(\theta) = V^{\pi_\theta}(s_{start})$. What assumption is made in this objective function?

Solution. We make the assumption that there is a single start state, s_{start}. In general, there can be a distribution of start states, in which case there should be an expectation over the distribution of start states, μ. Hence, the more general objective function is $J(\theta) = \mathbb{E}_{s_{start}\sim\mu}[V^{\pi_\theta}(s_{start})]$

Exercise 6.2. In the infinite horizon setting, we discussed in this set of lecture notes that a possible objective function is $J(\theta) = \mathbb{E}_{(s,a)\sim P_\theta(s,a)}[r(s,a)]$. In the lecture slides, we discuss two different objectives. We could either use Average Value given by $J_{avV}(\theta) = \sum_s d^{\pi_\theta}(s)V^{\pi_\theta}(s)$, or Average Reward per Time Step given by $J_{avR}(\theta) = \sum_s d^{\pi_\theta}(s)\sum_a \pi_\theta(a|s)r(s,a)$. Is $J(\theta)$ equivalent to $J_{avV}(\theta)$ or $J_{avR}(\theta)$ or neither?

Solution. $J(\theta)$ is equivalent to Average Reward per Time Step. In particular, the expectation over (s,a) drawn from $P_\theta(s,a)$ can be expanded into an expectation over s drawn from stationary state distribution $d^{\pi_\theta}(s)$ and an expectation over a drawn from policy $\pi_\theta(a|s)$.

Exercise 6.3. What is the key advantage of using finite difference to estimate policy gradients?

Solution. This method works for arbitrary policies, even if the policy is not differentiable.

Exercise 6.4. What is the point of the log derivative trick in policy gradients?

Solution. The log derivative trick allows the gradient estimation to be independent of the dynamics model which, in general, is unknown.

Exercise 6.5. In the derivation to prove that baseline as a function of state is unbiased, we used the fact that $\mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t|s_t)] = 0$. Provide steps to show this result.

Solution.

$$\begin{aligned}\mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t|s_t)] &= \int_a \pi_\theta(a_t|s_t)\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}da \\ &= \nabla_\theta \int_a \pi_\theta(a_t|s_t)da = \nabla_\theta 1 = 0\end{aligned}$$

Exercise 6.6. Why can't we perform the following optimization directly?

$$\max_{\pi'} J(\pi') = \max_{\pi'} J(\pi') - J(\pi) = \max_{\pi'} \mathbb{E}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^\pi(s_t, a_t)\right]$$

Solution. We want to find π' but to do that we need to do rollouts using π'. This process is too slow. We need to use Importance Sampling.

Exercise 6.7. Here is some pseudocode to perform Maximum Likelihood Estimation using automatic differentiation for discrete action space.

```
logits = policy.predictions(states)
negative_likelihoods = tf.nn.softmax_cross_entropy_with_logits(
                        labels=actions, logits=logits)
loss = tf.reduce_mean(negative_likelihoods)
gradients = loss.gradients(loss, variables)
```

Given that we rollout N episodes, each with a horizon of T and there are d_a distinct actions and d_s state dimensions, what are the shapes of `actions` and `states`?

We are also given a tensor for `q_values`. What should the shape of this tensor be?

Given `q_values`, how would you change the above pseudocode to do policy gradient training?

Solution. The shape of `actions` should be $(N * T, d_a)$. The shape of `states` should be $(N * T, d_s)$. The shape of `q_values` should be $(N * T, 1)$.

```
logits = policy.predictions(states)
negative_likelihoods = tf.nn.softmax_cross_entropy_with_logits(
                        labels=actions, logits=logits)
weighted_negative_likelihoods = tf.multiply(negative_likelihoods, q_values)
loss = tf.reduce_mean(weighted_negative_likelihoods)
gradients = loss.gradients(loss, variables)
```

Hence, Policy Gradient can be viewed as a form of weighted MLE where the weight is the expected discounted return of taking action a from state s. The higher the expected discounted return, the higher the weight, the larger the gradient and hence the bigger the update.

References

[1] http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-5.pdf

[2] http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-9.pdf

[3] Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S., Abbeel, P. (2018). Variance reduction for policy gradient with action-dependent factorized baselines. arXiv preprint arXiv:1803.07246.

[4] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization.arXiv preprint arXiv:1705.10528, 2017.

[5] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region-policy optimization. InInternational Conference on Machine Learning, pages 1889–1897,2015.

A TRPO Proofs

A.1 Reward Shaping

Here we provide a proof for Lemma 5.1.

Proof.

$$\begin{aligned} d^\pi &= (1-\gamma)(I-\gamma P_\pi)^{-1}\mu \\ (I-\gamma P_\pi)d^\pi &= (1-\gamma)\mu \end{aligned}$$

Now, by taking a dot product with $f(s)$, we have

$$\mathop{\mathbb{E}}_{s\sim d^\pi}[f(s)] - \mathop{\mathbb{E}}_{\substack{s\sim d^\pi \\ a\sim\pi(\cdot|s) \\ s'\sim M(\cdot|s,a)}}[\gamma f(s')] = (1-\gamma)\mathop{\mathbb{E}}_{s\sim\mu}[f(s)] \tag{11}$$

Rearranging the terms completes the proof. □

A.2 Bounding Difference in State Distributions

Here we provide a proof for Lemma 5.2.

Proof. Recall from (3) that $d^\pi = (1-\gamma)(I-\gamma P_\pi)^{-1}\mu$.

Define $G = (I-\gamma P_\pi)^{-1}$, $\bar{G} = (I-\gamma P_{\pi'})^{-1}$, and $\Delta = P_{\pi'} - P_\pi$.

We have

$$\begin{aligned} G^{-1} - \bar{G}^{-1} &= (I-\gamma P_\pi) - (I-\gamma P_{\pi'}) \\ &= \gamma(P_{\pi'} - P_\pi) \\ &= \gamma\Delta \\ \Rightarrow \bar{G} - G &= \bar{G}(G^{-1} - \bar{G}^{-1})G = \gamma\bar{G}\Delta G \end{aligned}$$

This allows us to derive

$$\begin{aligned} d^{\pi'} - d^\pi &= (1-\gamma)(\bar{G} - G)\mu \\ &= (1-\gamma)\gamma\bar{G}\Delta G\mu \\ &= \gamma\bar{G}\Delta(1-\gamma)G\mu \\ &= \gamma\bar{G}\Delta d^\pi \end{aligned} \tag{12}$$

Taking the l_1-norm of (12), we have by property of operator norm

$$\left\|d^{\pi'} - d^\pi\right\|_1 = \gamma\left\|\bar{G}\Delta d^\pi\right\|_1 \le \gamma\left\|\bar{G}\right\|_1 \|\Delta d^\pi\|_1 \tag{13}$$

Let us first bound $\left\|\bar{G}\right\|_1$.

$$\left\|\bar{G}\right\|_1 = \left\|(I-\gamma P_\pi)^{-1}\right\|_1 = \left\|\sum_{t=0}^{\infty}\gamma^t P_\pi^t\right\|_1 \le \sum_{t=0}^{\infty}\gamma^t \|P_\pi\|_1^t = \frac{1}{1-\gamma} \tag{14}$$

We are left with bounding $\|\Delta d^\pi\|_1$.

$$
\begin{aligned}
\|\Delta d^\pi\|_1 &= \sum_{s'} \left| \sum_s \Delta(s'|s) d^\pi(s) \right| \\
&\leq \sum_{s',s} |\Delta(s'|s)| d^\pi(s) \\
&= \sum_{s',s} \left| \sum_a \left(M(s'|s,a)\pi'(a|s) - M(s'|s,a)\pi(a|s) \right) \right| d^\pi(s) \\
&= \sum_{s',s} \left| \sum_a (M(s'|s,a)(\pi'(a|s) - \pi(a|s)) \right| d^\pi(s) \\
&\leq \sum_{s',a,s} M(s'|s,a) \left| \pi'(a|s) - \pi(a|s) \right| d^\pi(s) \\
&= \sum_{s,a} \left| \pi'(a|s) - \pi(a|s) \right| d^\pi(s) \\
&= \sum_s d^\pi(s) \sum_a |\pi'(a|s) - \pi(a|s)| \\
&= 2 \mathop{\mathbb{E}}_{s \sim d^\pi} [D_{TV}(\pi' \| \pi)[s]] \qquad (15)
\end{aligned}
$$

Combining (13), (14) and (15), we have:

$$
\left\| d^{\pi'} - d^\pi \right\|_1 \leq \frac{2\gamma}{1-\gamma} \left[\mathop{\mathbb{E}}_{s \sim d^\pi} [D_{TV}(\pi' \| \pi)[s]] \right] \qquad (16)
$$

as desired. □

A.3 Bounding Difference in Returns

Here we provide a proof for Lemma 5.3.

Proof. Define $\delta_f(s,a,s') = R(s,a,s') + \gamma f(s') - f(s)$.

Since $V^\pi = \frac{1}{1-\gamma} \left(\mathop{\mathbb{E}}_{\substack{s \sim d^\pi \\ a \sim \pi(\cdot|s) \\ s' \sim M(\cdot|s,a)}} [\delta_f(s,a,s')] \right) + \mathbb{E}_{s \sim \mu}[f(s)]$, we have:

$$
V^{\pi'} - V^\pi = \frac{1}{1-\gamma} \left(\mathop{\mathbb{E}}_{\substack{s \sim d^{\pi'} \\ a \sim \pi'(\cdot|s) \\ s' \sim M(\cdot|s,a)}} [\delta_f(s,a,s')] - \mathop{\mathbb{E}}_{\substack{s \sim d^\pi \\ a \sim \pi(\cdot|s) \\ s' \sim M(\cdot|s,a)}} [\delta_f(s,a,s')] \right) \qquad (17)
$$

Let us first focus on the first term.

Let $\bar{\delta_f}^{\pi'} \in \mathbb{R}^{|S|}$ where $\bar{\delta_f}^{\pi'}(s) = \mathop{\mathbb{E}}_{\substack{a \sim \pi'(\cdot|s) \\ s' \sim M(\cdot|s,a)}} [\delta_f(s,a,s')]$.

Now we derive an upper bound

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim M}}[\delta_f(s,a,s')] &= \left\langle d^{\pi'}, \bar{\delta_f}^{\pi'} \right\rangle \\
&= \left\langle d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle + \left\langle d^{\pi'} - d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle \\
&\leq \left\langle d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle + \left\| d^{\pi'} - d^{\pi} \right\|_1 \left\| \bar{\delta_f}^{\pi'} \right\|_\infty \\
&= \left\langle d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle + \left\| d^{\pi'} - d^{\pi} \right\|_1 \epsilon_f^{\pi'} \qquad (18)
\end{aligned}$$

where $\epsilon_f^{\pi'} = \max_s \left[\mathop{\mathbb{E}}_{\substack{a\sim\pi'(\cdot|s)\\ s'\sim M(\cdot|s,a)}} [R(s,a,s') + \gamma f(s') - f(s)] \right]$.

We also have a lower bound

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim M}}[\delta_f(s,a,s')] &= \left\langle d^{\pi'}, \bar{\delta_f}^{\pi'} \right\rangle \\
&= \left\langle d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle - \left\langle d^{\pi} - d^{\pi'}, \bar{\delta_f}^{\pi'} \right\rangle \\
&\geq \left\langle d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle - \left\| d^{\pi} - d^{\pi'} \right\|_1 \left\| \bar{\delta_f}^{\pi'} \right\|_\infty \\
&= \left\langle d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle - \left\| d^{\pi'} - d^{\pi} \right\|_1 \epsilon_f^{\pi'} \qquad (19)
\end{aligned}$$

Now, we apply (18) to (17) to get an upper bound:

$$\begin{aligned}
(1-\gamma)(V^{\pi'} - V^{\pi}) &\leq \left\langle d^{\pi}, \bar{\delta_f}^{\pi'} \right\rangle + \left\| d^{\pi'} - d^{\pi} \right\|_1 \epsilon_f^{\pi'} - \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)\\ s'\sim M(\cdot|s,a)}} [\delta_f(s,a,s')] \\
&= \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'(\cdot|s)\\ s'\sim M(\cdot|s,a)}} [\delta_f(s,a,s')] - \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)\\ s'\sim M(\cdot|s,a)}} [\delta_f(s,a,s')] + \left\| d^{\pi'} - d^{\pi} \right\|_1 \epsilon_f^{\pi'} \\
&= \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)\\ s'\sim M(\cdot|s,a)}} \left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1 \right) \delta_f(s,a,s') \right] + \left\| d^{\pi'} - d^{\pi} \right\|_1 \epsilon_f^{\pi'} \qquad (20)
\end{aligned}$$

Let us define

$$L_\pi(\pi') = \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)\\ s'\sim M(\cdot|s,a)}} \left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1 \right) (R(s,a,s') + \gamma f(s') - f(s)) \right]$$

Note that if we select $f = V^\pi$, then

$$L_\pi(\pi') = \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^\pi(s,a) \right]$$

This is because the advantage of choosing an action from the same policy is 0.

$$\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)}} [A^\pi(s,a)] = 0$$

We complete (20) to obtain an upper bound:

$$V^{\pi'} - V^{\pi} \leq \frac{1}{1-\gamma}\left(L_{\pi}(\pi') + \left\|d^{\pi'} - d^{\pi}\right\|_1 \epsilon_f^{\pi'}\right) \tag{21}$$

Similarly, we derive a lower bound from (17) and (19).

$$\begin{aligned} V^{\pi'} - V^{\pi} &\geq \frac{1}{1-\gamma}\left(\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi(\cdot|s)\\ s'\sim M(\cdot|s,a)}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1\right)\delta_f(s,a,s')\right] - \left\|d^{\pi'} - d^{\pi}\right\|_1 \epsilon_f^{\pi'}\right) \\ &= \frac{1}{1-\gamma}\left(L_{\pi}(\pi') - \left\|d^{\pi'} - d^{\pi}\right\|_1 \epsilon_f^{\pi'}\right) \end{aligned} \tag{22}$$

□

A.4 Maximum Advantage

Here we provide a proof of Lemma 5.4.

Proof. As in Section A of [5], we say (π, π') is an α-coupled policy pair if it defines a joint distribution $(a, \hat{a}) \mid s$, such that $\forall s, \Pr[a \neq \hat{a} \mid s] \leq \alpha$. Define $\alpha_{\pi,\pi'}$ such that (π, π') are $\alpha_{\pi,\pi'}$-coupled. Let $\bar{A}(s)$ be the expected value of $A^{\pi}(s, \hat{a})$ under the expectation of $\hat{a}$ drawn from policy π' given state s.

$$\bar{A}(s) = \mathop{\mathbb{E}}_{\hat{a}\sim\pi'(\cdot|s)}\left[A^{\pi}(s,\hat{a})\right] \tag{23}$$

Because $\mathbb{E}_{a\sim\pi(\cdot|s)}\left[A^{\pi}(s,a) \mid s\right] = 0$ from the definition of advantage, we rewrite (23) as

$$\begin{aligned} \bar{A}(s) &= \mathop{\mathbb{E}}_{(a,\hat{a})\sim(\pi,\pi')}\left[A^{\pi}(s,\hat{a}) - A^{\pi}(s,a)\right] \\ &= \Pr\left[a \neq \hat{a} \mid s\right] \mathop{\mathbb{E}}_{(a,\hat{a})\sim(\pi,\pi')}\left[A^{\pi}(s,\hat{a}) - A^{\pi}(s,a) \mid a \neq \hat{a}\right] \\ &+ \Pr\left[a = \hat{a} \mid s\right] \mathop{\mathbb{E}}_{(a,\hat{a})\sim(\pi,\pi')}\left[A^{\pi}(s,a) - A^{\pi}(s,a)\right] \\ &\leq \alpha_{\pi,\pi'} \mathop{\mathbb{E}}_{(a,\hat{a})\sim(\pi,\pi')}\left[A^{\pi}(s,\hat{a}) - A^{\pi}(s,a) \mid a \neq \hat{a}\right] + \Pr\left[a = \hat{a} \mid s\right] * 0 \\ &\leq 2\alpha_{\pi,\pi'} \max_a \left|A^{\pi}(s,a)\right| \end{aligned} \tag{24}$$

Therefore, we have

$$\begin{aligned} \epsilon_{V^{\pi}}^{\pi'} &= \max_s \left| \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'(\cdot|s)\\ s'\sim M(\cdot|s,a)}}\left[R(s,a,s') + \gamma f(s') - f(s)\right] \right| \\ &\leq \max_s \left| 2\alpha_{\pi,\pi'} \max_a \left|A^{\pi}(s,a)\right| \right| \\ &\leq 2\alpha_{\pi,\pi'} \max_{s,a} \left|A^{\pi}(s,a)\right| \end{aligned} \tag{25}$$

Suppose p_X and p_Y are distributions with $D_{TV}(p_X \| p_Y) = \alpha$, then there exists a joint distribution (X, Y) whose marginals are p_X, p_Y for which $X = Y$ with probability $1 - \alpha$ [5]. Taking $\alpha_{\pi,\pi'} = \max_s D_{TV}(\pi \| \pi')$ we have

$$\epsilon_{V^\pi}^{\pi'} \leq 2 \max_s D_{TV}(\pi \| \pi') \max_{s,a} |A^\pi(s,a)| \tag{26}$$

as desired. □