Bridging the Gap Between Value and Policy Based Reinforcement Learning

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, Dale Schuurmans

Topic: Q-Value Based RL
Presenter: Michael Pham-Hung
Motivation
Motivation

Value Based RL

- Data efficient
- Learn from any trajectory

i.e. Q-Learning
Motivation

Value Based RL

+ Data efficient
+ Learn from any trajectory

Policy Based RL

i.e. REINFORCE

+ Stable deep function approximators

i.e. Q-Learning

+ Data efficient
+ Learn from any trajectory
Motivation

Value Based RL

- i.e. Q-Learning
- + Data efficient
- + Learn from any trajectory

Policy Based RL

- i.e. REINFORCE
- + Stable deep function approximators
Motivation

Value Based RL

- Data efficient
- Learn from any trajectory

Policy Based RL

- Stable deep function approximators

Profit.

i.e. Q-Learning

i.e. REINFORCE
Contributions

Problem: Combining the advantages of on-policy and off-policy learning.
Contributions

Problem: Combining the advantages of on-policy and off-policy learning.

Why is this problem important?:

- Model-free RL with deep function approximators seems like a good idea.
Contributions

Problem: Combining the advantages of on-policy and off-policy learning.

Why is this problem important?:
- Model-free RL with deep function approximators seems like a good idea.

Why is this problem hard?:
- Value-based learning is not always stable with deep function approximators.
Contributions

**Problem**: Combining the advantages of on-policy and off-policy learning.

**Why is this problem important?**:
- Model-free RL with deep function approximators seems like a good idea.

**Why is this problem hard?**:
- Value-based learning is not always stable with deep function approximators.

**Limitations of prior work**:
- Prior work remain potentially unstable and are not generalizable.
Contributions

Problem: Combining the advantages of on-policy and off-policy learning.

Why is this problem important?:
- Model-free RL with deep function approximators seems like a good idea.

Why is this problem hard?:
- Value-based learning is not always stable with deep function approximators.

Limitations of prior work:
- Prior work remain potentially unstable and are not generalizable.

Key Insight: Starting from first principles rather than performing naïve approaches can be more rewarding.

Revealed: Results in flexible algorithm.
Outline

Background
- Q-Learning Formulation
- Softmax Temporal Consistency
- Consistency between optimal value and policy

PCL Algorithm
- Basic PCL
- Unified PCL

Results

Limitations
Q-Learning Formulation

\[ Q_{ER}(s, \pi) = \sum_a \pi(a|s)[r(s, a) + \gamma(Q_{ER}(s', \pi))], \text{ where } s' = f(s, a) \]
Q-Learning Formulation

\[ O_{ER}(s, \pi) = \sum_a \pi(a|s)[r(s, a) + \gamma O_{ER}(s', \pi)], \text{ where } s' = f(s, a) \]

\[ V^*(s) = \max_{\pi} O_{ER}(s, \pi), \quad \pi^o = \arg\max_{\pi} O_{ER}(s, \pi) \]
Q-Learning Formulation

$$O_{ER}(s, \pi) = \sum_a \pi(a|s)[r(s,a) + \gamma O_{ER}(s',\pi)], \text{ where } s' = f(s,a)$$

$$V^\circ(s) = \max_\pi O_{ER}(s,\pi), \quad \pi^o = \arg\max_\pi O_{ER}(s,\pi)$$
Q-Learning Formulation

\[ O_{ER}(s, \pi) = \sum_{a} \pi(a|s)[r(s, a) + \gamma O_{ER}(s', \pi)], \text{ where } s' = f(s, a) \]

\[ V^\circ(s) = \max_{\pi} O_{ER}(s, \pi), \quad \pi^\circ = \arg\max_{\pi} O_{ER}(s, \pi) \]

\[ V^\circ(s) = O_{ER}(s, \pi^\circ) = \max_{a}(r(s, a) + \gamma V^\circ(s')) \]
Q-Learning Formulation

\[ O_{\text{ER}}(s, \pi) = \sum_a \pi(a|s)(r(s, a) + \gamma O_{\text{ER}}(s', \pi)), \text{ where } s' = f(s, a) \]

\[ V^*(s) = \max_{\pi} O_{\text{ER}}(s, \pi), \quad \pi^* = \arg\max_{\pi} O_{\text{ER}}(s, \pi) \]

\[ V^*(s) = O_{\text{ER}}(s, \pi^*) = \max_a (r(s, a) + \gamma V^*(s')) \]

Hard-max Bellman temporal consistency!
Q-Learning Formulation

\[ O_{ER}(s, \pi) = \sum_a \pi(a|s)[r(s, a) + \gamma O_{ER}(s', \pi)], \text{ where } s' = f(s, a) \]

\[ V^\circ(s) = \max_\pi O_{ER}(s, \pi), \quad \pi^\circ = \arg\max_\pi O_{ER}(s, \pi) \]

\[ V^\circ(s) = O_{ER}(s, \pi^\circ) = \max_a (r(s, a) + \gamma V^\circ(s')) \]

Or in terms of optimal action values

\[ Q^\circ(s, a) = r(s, a) + \gamma \max_{a'} Q^\circ(s', a') \]
Soft-max Temporal Consistency

- Augment the standard expected reward objective with a discounted entropy regularizer

- This helps encourages exploration and helps prevent early convergence to sub-optimal policies
\[ O_{\text{ENT}}(s, \pi) = O_{\text{ER}}(s, \pi) + \tau \mathbb{H}(s, \pi) \]
\[ O_{\text{ENT}}(s, \pi) = O_{\text{ER}}(s, \pi) + \tau \mathbb{H}(s, \pi) \]

\[ \mathbb{H}(s, \pi) = \sum_a \pi(a|s)[-\log \pi(a|s) + \gamma \mathbb{H}(s', \pi)] \]
\[ O_{ENT}(s, \pi) = O_{ER(s, \pi)} + \tau \mathbb{H}(s, \pi) \]

\[ \mathbb{H}(s, \pi) = \sum_{a} \pi(a|s) \left( -\log \pi(a|s) + \gamma \mathbb{H}(s', \pi) \right) \]

\[ O_{ENT}(s, \pi) = \sum_{a} \pi(a|s) \left( r(s, a) - \tau \log \pi(a|s) + \gamma O_{ENT}(s', \pi) \right) \]
\[ O_{ENT}(s, \pi) = O_{ER(s,\pi)} + \tau \mathcal{H}(s, \pi) \]

\[ \mathcal{H}(s, \pi) = \sum_a \pi(a|s)[-\log \pi(a|s) + \gamma \mathcal{H}(s', \pi)] \]

\[ O_{ENT}(s, \pi) = \sum_a \pi(a|s)[r(s, a) - \tau \log \pi(a|s) + \gamma O_{ENT}(s', \pi)] \]

\[ \pi^*(a|s) \propto \exp \left\{ \frac{r(s, a) + \gamma V^*(s')}{\tau} \right\} \]

Form of a Boltzmann distribution … No longer one hot distribution! Entropy term prefers the use of policies with more uncertainty.
Subbing $\pi^*(a|s)$ into $O_{ENT}$ yields:

$$V^*(s) = O_{ENT}(s, \pi^*) = \tau \log \sum_a \exp \left\{ \frac{r(s, a) + \gamma V^*(s')}{\tau} \right\}$$

And

$$Q^*(s, a) = r(s, a) + \gamma V^*(s') = r(s, a) + \gamma \tau \log \sum_{a'} \exp \left( \frac{Q^*(s', a')}{\tau} \right)$$
Subbing $\pi^*(a|s)$ into $O_{ENT}$ yields:

$$V^*(s) = O_{ENT}(s, \pi^*) = \tau \log \sum_a \exp \left\{ \frac{r(s, a) + \gamma V^*(s')}{\tau} \right\}$$

And

$$Q^*(s, a) = r(s, a) + \gamma V^*(s') = r(s, a) + \gamma \tau \log \sum_{a'} \exp \left( \frac{Q^*(s', a')}{\tau} \right)$$

Note the log-sum-exp form!
Consistency Between Optimal Value & Policy

• Let \( \pi^*(a|s) = \frac{\exp\{(r(s,a) + \gamma V^*(s'))/\tau\}}{\exp\{V^*(s)/\tau\}} \) — Normalization Factor
Consistency Between Optimal Value & Policy

- Let \( \pi^*(a|s) = \frac{\exp\left\{\frac{(r(s,a) + \gamma V^*(s'))}{\tau}\right\}}{\exp\{V^*(s)/\tau\}} \)

  \[
  \log(\pi^*(a|s)) = \frac{(r(s, a) + \gamma V^*(s'))}{\tau} - \frac{V^*(s)}{\tau}
  \]
Consistency Between Optimal Value & Policy

- Let $\pi^*(a|s) = \frac{\exp\left\{ \frac{(r(s,a) + \gamma V^*(s'))}{\tau} \right\}}{\exp\{V^*(s)/\tau\}}$

$$\log(\pi^*(a|s)) = \frac{(r(s,a) + \gamma V^*(s'))}{\tau} - \frac{V^*(s)}{\tau}$$

$$V^*(s) - \gamma V^*(s') = r(s,a) - \tau \log \pi^*(a|s)$$
Consistency Between Optimal Value & Policy

- Let $\pi^*(a|s) = \frac{\exp\{r(s,a) + \gamma V^*(s')\}/\tau}{\exp\{V^*(s)/\tau\}}$

  $$\log(\pi^*(a|s)) = \frac{(r(s,a) + \gamma V^*(s'))}{\tau} - \frac{V^*(s)}{\tau}$$

  $$V^*(s) - \gamma V^*(s') = r(s,a) - \tau \log \pi^*(a|s)$$

  Valid for any action $a$
Consistency Between Optimal Value & Policy

Note: The optimal policy can also be characterized in terms of $Q^*$:

$$\pi^*(a|s) = \exp\left\{\frac{Q^*(s, a) - V^*(s)}{\tau}\right\}$$
Theorem 1: For \( \tau > 0 \), the policy \( \pi^* \) that maximizes \( O_{\text{ENT}} \) and state values \( V^*(s) = \max_{\pi} O_{\text{ENT}}(s, \pi) \) satisfy the following temporal consistency property for any state \( s \) and action \( a \) (where \( s' = f(s, a) \)):

\[
V^*(s) - \gamma V^*(s') = r(s, a) - \tau \log \pi^*(a|s)
\]
Theorem 1: For $\tau > 0$, the policy $\pi^*$ that maximizes $O_{\text{ENT}}$ and state values $V^*(s) = \max_\pi O_{\text{ENT}}(s, \pi)$ satisfy the following temporal consistency property for any state $s$ and action $a$ (where $s' = f(s, a)$):

$$V^*(s) - \gamma V^*(s') = r(s, a) - \tau \log \pi^*(a|s)$$

Corollary 2: For $\tau > 0$, the optimal policy $\pi^*$ and optimal state values $V^*$ satisfy the following extended temporal consistency property, for any state $s_1$ and any action sequence $a_1, ..., a_{t-1}$ (where $s_{i+1} = f(s_i, a_i)$):

$$V^*(s) - \gamma^{t-1} V^*(s') = \sum_{i=1}^{t-1} \gamma^{i-1} [r(s_i, a_i) - \tau \log \pi^*(a_i|s_i)]$$
**Theorem 1:** For $\tau > 0$, the policy $\pi^*$ that maximizes $O_{ENT}$ and state values $V^*(s) = \max_{\pi} O_{ENT}(s, \pi)$ satisfy the following temporal consistency property for any state $s$ and action $a$ (where $s' = f(s, a)$):

$$V^*(s) - \gamma V^*(s') = r(s, a) - \tau \log \pi^*(a|s)$$

**Corollary 2:** For $\tau > 0$, the optimal policy $\pi^*$ and optimal state values $V^*$ satisfy the following extended temporal consistency property, for any state $s_1$ and any action sequence $a_1, ..., a_{t-1}$ (where $s_{i+1} = f(s_i, a_i)$):

$$V^*(s) - \gamma^{t-1} V^*(s') = \sum_{i=1}^{t-1} \gamma^{i-1} [r(s_i, a_i) - \tau \log \pi^*(a_i|s_i)]$$

**Theorem 3.** If a policy $\pi(a | s)$ and state value function $V(s)$ satisfy the consistency theorem 1 for all states $s$ and actions $a$ (where $s' = f(s, a)$), then $\pi = \pi^*$ and $V = V^*$
Theorem 1: For $\tau > 0$, the policy $\pi^*$ that maximizes $O_{\text{ENT}}$ and state values $V^*(s)$
$= \max_{\pi} O_{\text{ENT}}(s, \pi)$ satisfy the following temporal consistency property for any state $s$ and action $a$
(where $s' = f(s, a)$):

$$V^*(s) - \gamma V^*(s') = r(s, a) - \tau \log \pi^*(a|s)$$

Corollary 2: For $\tau > 0$, the optimal policy $\pi^*$ and optimal state values $V^*$ satisfy the following
extended temporal consistency property, for any state $s_1$ and any action sequence $a_1, \ldots, a_{t-1}$
(where $s_{i+1} = f(s_i, a_i)$):

$$V^*(s) - \gamma V^*(s') = \sum_{i=1}^{t-1} \gamma^{i-1} [r(s_i, a_i) - \tau \log \pi^*(a_i|s_i)]$$

Theorem 3. If a policy $\pi(a \mid s)$ and state value function $V(s)$ satisfy the consistency theorem 1 for
all states $s$ and actions $a$ (where $s' = f(s, a)$), then $\pi = \pi^*$ and $V = V^*$
Algorithm - Path Consistency Learning (PCL)

PCL attempts to minimize the squared soft consistency error over a set of sub-trajectories $E$.

Define a notion of soft consistency for $d$-length sub-trajectory $s_{i:i+d} \equiv (s_i, a_i, ..., s_{i+d-1}, a_{i+d-1}, s_{i+d})$. From corollary 2.

$$C(s_{i:i+d}, \theta, \phi) = V_\phi(s_i) - \gamma^d V(s_{i+d}) + \sum_{j=0}^{d-1} \gamma^j [r(s_{i+j}, a_{i+j}) - \tau \log \pi_\theta(a_{i+j}|s_{i+j})]$$

Goal is to find $V_\phi$ and $\pi_\theta$ that gets $C(s_{i:i+d}, \theta, \phi)$ close to zero
Define squared soft consistency error:

\[ O_{PCL} = \sum_{s_i:i+d \in E} \frac{1}{2} C(s_{i:i+d}, \theta, \phi)^2 \]

We then get updates for \( \theta \) and \( \phi \) by taking the gradient:

\[ \Delta \theta = \eta_\pi C(s_{i:i+d}, \theta, \phi) \sum_{j=0}^{d-1} \gamma^j \nabla_\theta \log \pi_\theta(a_{i+j} \mid s_{i+j}) \]

\[ \Delta \phi = \eta_\nu C(s_{i:i+d}, \theta, \phi) [\nabla_\phi V_\phi(s_i) - \gamma^d \nabla_\phi V_\phi(s_{i+d})] \]

Where \( \eta_\pi \) and \( \eta_\nu \) are the learning rates for policy and value.
Algorithm - Path Consistency Learning (PCL)

Given that the consistency property must hold on any path, the PCL algorithm can apply the updates both to trajectories sampled on-policy from $\pi_\theta$ as well as trajectories sampled from a replay buffer.
Algorithm 1 Path Consistency Learning

**Input:** Environment \( ENV \), learning rates \( \eta_\pi, \eta_\nu \), discount factor \( \gamma \), rollout \( d \), number of steps \( N \), replay buffer capacity \( B \), prioritized replay hyperparameter \( \alpha \).

**function** Gradients\((s_{0:T})\)

// We use \( G(s_{t:t+d}, \pi_\theta) \) to denote a discounted sum of log-probabilities from \( s_t \) to \( s_{t+d} \).

- Compute \( \Delta\theta = \sum_{t=0}^{T-d} C_{\theta,\phi}(s_{t:t+d}) \nabla_\theta G(s_{t:t+d}, \pi_\theta) \).
- Compute \( \Delta\phi = \sum_{t=0}^{T-d} C_{\theta,\phi}(s_{t:t+d}) \left( \nabla_\phi V_\phi(s_t) - \gamma^d \nabla_\phi V_\phi(s_{t+d}) \right) \).

**Return** \( \Delta\theta, \Delta\phi \)

**end function**

Initialize \( \theta, \phi \).

Initialize empty replay buffer \( RB(\alpha) \).

for \( i = 0 \) to \( N - 1 \) do

- Sample \( s_{0:T} \sim \pi_\theta(s_0) \) on \( ENV \).
- \( \Delta\theta, \Delta\phi = \text{Gradients}(s_{0:T}) \).
- Update \( \theta \leftarrow \theta + \eta_\pi \Delta\theta \).
- Update \( \phi \leftarrow \phi + \eta_\nu \Delta\phi \).
- Input \( s_{0:T} \) into \( RB \) with priority \( R^1(s_{0:T}) \).
- If \( |RB| > B \), remove episodes uniformly at random.
- Sample \( s_{0:T} \) from \( RB \).
- \( \Delta\theta, \Delta\phi = \text{Gradients}(s_{0:T}) \).
- Update \( \theta \leftarrow \theta + \eta_\pi \Delta\theta \).
- Update \( \phi \leftarrow \phi + \eta_\nu \Delta\phi \).

**end for**
Algorithm - Unified PCL

Recall:

\[ Q^*(s, a) = r(s, a) + \gamma V^*(s') = r(s, a) + \gamma \tau \log \sum_{a'} \exp \left( \frac{Q^*(s', a')}{\tau} \right) \]

\[ \Rightarrow V_\rho (s) = \tau \log \sum_a \exp \left( \frac{Q_\rho (s, a)}{\tau} \right) \]

And,

\[ \pi_\rho (a|s) = \exp \left( \frac{(Q_\rho (s, a) - V_\rho (s))}{\tau} \right) \]
Algorithm - Unified PCL

Then the new update rule for $\rho$ is:

\[
\Delta \rho = \eta_c C(s_{i:i+d}, \rho) \sum_{j=0}^{d-1} \gamma^j \nabla_{\rho} \log \pi_{\rho}(a_{i+j} | s_{i+j}) \\
+ \eta_v C(s_{i:i+d}, \rho) \left( \nabla_{\rho} V_{\rho}(s_i) - \gamma^d \nabla_{\rho} V_{\rho}(s_{i+d}) \right)
\]

Merging the policy and value function models in this way is significant because it presents a new actor-critic paradigm where the policy (actor) is not distinct from the values (critic)
Experimental Results

PCL can consistently match or beat the performance of A3C and double Q-learning.

PCL and Unified PCL are easily implementable with expert trajectories. Expert trajectories can be prioritized in the replay buffer as well.
The results of PCL against A3C and DQN baselines. Each plot shows average reward across 5 random training runs (10 for Synthetic Tree) after choosing best hyperparameters. A signal standard deviation bar clipped at the min and max. The x-axis is number of training iterations. PCL exhibits comparable performance to A3C in some tasks, but clearly outperforms A3C on the more challenging tasks. Across all tasks, the performance of DQN is worse than PCL.
The results of PCL vs. Unified PCL. Overall found that using a single model for both values and policy is not detrimental to training. Although in some of the simpler tasks PCL has an edge over Unified PCL, on the more difficult tasks, Unified PCL performs better.
The results of PCL vs. PCL augmented with a small number of expert trajectories on the hardest algorithmic tasks. We find that incorporating expert trajectories greatly improves performance.
Discussion of results

Using a single model for both values and policy is not detrimental to training

The ability for PCL to incorporate expert trajectories without requiring adjustment or correction is a desirable property in real-world applications
Critique / Limitations / Open Issues

- Only implemented on simple tasks
  - Addressed with Trust-PCL, which enables a continuous action space.

- Requires small learning rates
  - Addressed with Trust-PCL, which uses Trust Regions

- Proof was given for deterministic states but also works for stochastic states as well.
Contributions

**Problem:** Combining the advantages of on-policy and off-policy learning.

**Why is this problem important?:**
- Model-free RL with deep functions approximators seems like a good idea.

**Why is this problem hard?:**
- Value-based learning is not always stable with deep function approximators.

**Limitations of prior work:**
- Prior work remain potentially unstable and are not generalizable.

**Key Insight:** Starting from a theoretical approach rather than naïve approaches can be more fruitful.

**Revealed:** Results in a quite flexible algorithm.
Exercise Questions

1. What are the advantages from Value-based Learning and what are the advantages from Policy-based Learning

2. Derive the Softmax temporal consistency from $\pi^* = \frac{\exp\left\{ \frac{(r(s,a)+\gamma V^*(s'))}{\tau} \right\}}{\exp\{V^*(s)/\tau\}}$