Operator World Models for Reinforcement Learning

Pietro Novelli
Joint work with M .Pratticò, M. Pontil and C. Ciliberto

Impact on the Real World

- Robotics and Autonomous Systems,
- Finance: Trading and Portfolio Opt.,
- Energy Management and Smart Grids,
- Healthcare and Personalized Treatment,
- Games and Decision-Making,
- Autonomous Vehicles,
- RLHF (RL w/ human feedback),
- Reasoning for LLMs,
- ...

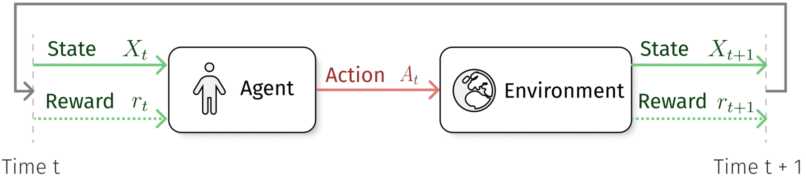
2024 Turing Award awarded to Sutton and Barto for laying the foundations of reinforcement learning.

Problem Setting

Intuition

We have a sequential decision making problem:

At time t, the environment is in the state X_t. An agent executes an action A_t. The environment changes its state to X_{t+1}.



A scalar reward $r_{t+1} = r(X_t, A_t)$ tells us how good/bad this was.

Our goal is to find a **policy** to choose the “best” actions.

Markov Decision Processes (MDPs)

We consider a Markov Decision Process (MDP) characterized by a:

- State space $\mathcal{X}$
- Action space $\mathcal{A}$
- A transition kernel $\tau : \Omega = \mathcal{X} \times \mathcal{A} \to \mathcal{P}(\mathcal{X})$
- A (non-negative) reward[1] $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}_+$

Details

We assume $\mathcal{X}$ and $\mathcal{A}$ Polish spaces and τ, r Borel measurable. $\mathcal{P}(\mathcal{X})$ the space of Borel probability measures on $\mathcal{X}$.

[1] Making this technically a Markov *Reward* Process.

Policy

We assume to have the freedom to choose what action to take.

This is embodied in the notion of a **policy:**

$$\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$$

so that $\pi(\cdot|x)$ denotes the probability that we will take a specific action when the system is in state $x \in \mathcal{X}$.

Goal. Find the “best” policy?

Objective

Given a starting distribution $\nu \in \mathcal{P}(\mathcal{X})$ and a discount[2] $\gamma \in [0, 1)$,

Goal: maximize the (γ-discounted) Expected Return

$$J(\pi) = \mathbb{E}_{\nu,\pi,\tau}\left[\sum_{t=0}^{+\infty} \gamma^t r(X_t, A_t)\right]$$

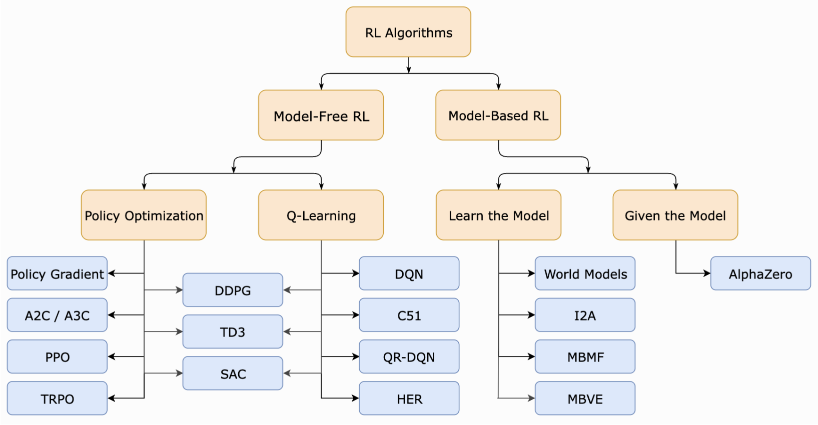
achieved by policy π in the MDP $(\mathcal{X}, \mathcal{A}, \tau, r)$ starting from ν.

Notation:

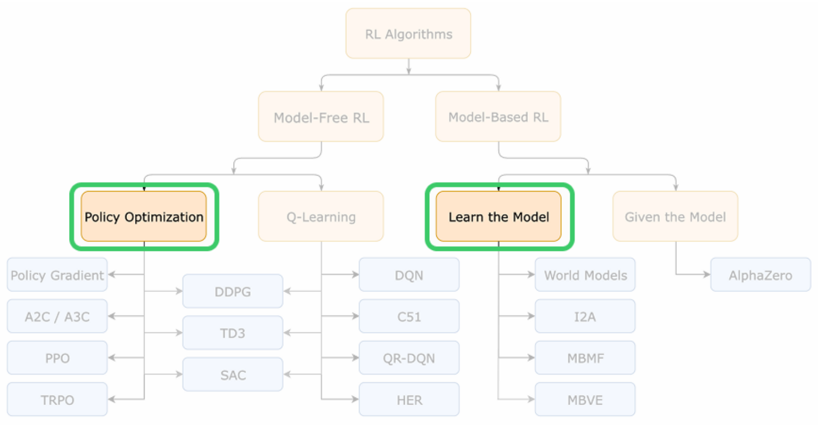
X_0, A_t and X_{t+1} have laws respectively ν, $\pi(\cdot|X_t)$ and $\tau(\cdot|X_t, A_t)$ for any $t \in \mathbb{N}$.

[2] this could be removed, but makes for much more manageable problems.

A Taxonomy of RL Algorithms



A Taxonomy of RL Algorithms



Today's plan

We will formalize the RL problem with an operatorial formalism through which we will

- Derive a coincise expression of the objective function, and its derivatives.
- Study the convergence rates of (policy) mirror descent.
- Describe an RL algorithm based on conditional mean embeddings, if time permits.

Operator-based Formulation

Operatorial Perspective

Transfer Operator.

$\mathsf{T} : B_b(\mathcal{X}) \to B_b(\Omega)$ such that for any $f \in B_b(\mathcal{X})$ and $(x, a) \in \Omega$

$$(\mathsf{T}f)(x,a) = \int_{\mathcal{X}} f(x') \, \tau(dx'|x,a) = \mathbb{E}\left[f(X') \mid x, a\right].$$

"Policy" Operator.

$\mathsf{P}_\pi : B_b(\Omega) \to B_b(\mathcal{X})$ such that for any $g \in B_b(\Omega)$ and $x \in \mathcal{X}$

$$(\mathsf{P}_\pi g)(x) = \int_{\mathcal{A}} g(x,a) \, \pi(da|x) = \mathbb{E}\left[g(X,A) \mid X = x\right].$$

Notation:

$B_b(\mathcal{X})$ space of bounded Borel-measurable functions on $\mathcal{X}$.

Markov Operators

Both T and P_π are Markov operators.

$$f \geq 0 \implies \mathsf{M}f \geq 0$$
$$\mathsf{M}1 = 1$$

Every Markov operator is associated to a conditional probability $p(\cdot|x) = \mathsf{M}^*\delta_x$ through its adjoint.

Markov operators are a convex subset of $B_b(\mathcal{X})$ and they all have norm $\|\mathsf{M}\| = 1$.

Why on earth?

- With operators we replace the non-linear evolution law $\tau(x'|x,a)$ for the linear law $f \mapsto \mathsf{T}f$.
- T describes expected values in the future, which are exactly what appears in the RL objective function.
- A non-standard approach to RL.

Operatorial Perspective (II)

By applying T and P_π we have...

Single interaction between π and the MDP starting from (x, a):

$$\mathbb{E}[r(X_1, A_1)|X_0 = x, A_0 = a] = (\mathsf{T}\mathsf{P}_\pi r)(x, a)$$

Operatorial Perspective (II)

By applying T and P_π we have...

Single interaction between π and the MDP starting from (x, a):

$$\mathbb{E}[r(X_1, A_1) | X_0 = x, A_0 = a] = (\mathsf{T}\mathsf{P}_\pi r)(x, a)$$

After $t \in \mathbb{N}$ such steps...

$$\mathbb{E}[r(X_t, A_t) | X_0 = x, A_0 = a] = \left[(\mathsf{T}\mathsf{P}_\pi)^t r\right](x, a)$$

Operatorial Perspective (II)

By applying T and P_π we have...

Single interaction between π and the MDP starting from (x, a):

$$\mathbb{E}[r(X_1, A_1) | X_0 = x, A_0 = a] = (\mathsf{T}\mathsf{P}_\pi r)(x, a)$$

After $t \in \mathbb{N}$ such steps...

$$\mathbb{E}[r(X_t, A_t) | X_0 = x, A_0 = a] = \left[(\mathsf{T}\mathsf{P}_\pi)^t r\right](x, a)$$

Summing all of them up (with γ-discount):

$$q_\pi(x, a) = \sum_{t=0}^{+\infty} \gamma^t \mathbb{E}[r(X_t, A_t) | x, a] = \sum_{t=0}^{+\infty} (\gamma \mathsf{T}\mathsf{P}_\pi)^t r = (\mathsf{Id} - \gamma \mathsf{T}\mathsf{P}_\pi)^{-1} r(x, a)$$

Why: T and P_π are Markov operators, hence their operator norms $\|\mathsf{T}\| = \|\mathsf{P}_\pi\| = 1$ and therefore the above Neumann series is convergent.

Operatorial Perspective (III)

Proposition.[3]

$$J(\pi) = \langle \mathsf{P}_\pi q_\pi, \nu \rangle = \left\langle \mathsf{P}_\pi (\mathsf{Id} - \gamma \mathsf{T} \mathsf{P}_\pi)^{-1} r, \nu \right\rangle$$

[3] With the canonical pairing $\langle f, \nu \rangle = \int f(x) \nu(dx)$.

Operatorial Perspective (III)

Proposition.[3]

$$J(\pi) = \langle \mathsf{P}_\pi q_\pi, \nu \rangle = \left\langle \mathsf{P}_\pi (\mathsf{Id} - \gamma \mathsf{T}\mathsf{P}_\pi)^{-1} r, \nu \right\rangle$$

Proof. Recall the definition of our objective

$$\begin{aligned}
J(\pi) &= \mathbb{E}_{\nu,\pi,\tau}\left[\sum_{t=0}^{+\infty} \gamma^t r(X_t, A_t)\right] \\
&= \mathbb{E}_{\nu,\pi}\left[\sum_{t=0}^{+\infty} \gamma^t \mathbb{E}[r(X_t, A_t)|X_0, A_0]\right] \\
&= \mathbb{E}_{\nu,\pi}\left[q_\pi(X_0, A_0)\right] \\
&= \mathbb{E}_{\nu}\left[\mathbb{E}_{\pi(\cdot|X_0)}[q_\pi(X_0, A_0)|X_0]\right] \\
&= \mathbb{E}_{\nu}\left[(\mathsf{P}_\pi q_\pi)(X_0)\right] \\
&= \langle \mathsf{P}_\pi q_\pi, \nu \rangle
\end{aligned}$$

[3] With the canonical pairing $\langle f, \nu \rangle = \int f(x)\nu(dx)$.

Minimizing J

Why do we like the operator form for $J(\pi)$? Well, for starters,

$$\max_{\pi} \left\langle \mathsf{P}_\pi (\mathsf{Id} - \gamma \mathsf{T} \mathsf{P}_\pi)^{-1} r, \nu \right\rangle$$

is in a much more "standard" form (from an optimization perspective).

Actually... since for any $\theta \in [0,1]$ and policies π_1, π_2,

$$\mathsf{P}_{\theta\pi_1 + (1-\theta)\pi_2} = \theta \mathsf{P}_{\pi_1} + (1-\theta)\mathsf{P}_{\pi_2}$$

The definition of policy operator is **linear** w.r.t. individual policies, so...

...is the problem convex? (or rather, concave, since it's a maximization?)

Minimizing J

Unfortunately, not[4].

[4] Agarwal, Kakade, Lee and Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. JMLR 2021.

[5] We have generalized the definitions of J and q to any Markov operator.

Minimizing J

Unfortunately, not[4].

However, nothing prevents try and minimize it nevertheless!

In particular, since we can "easily" compute derivatives…

Lemma.[5] For any Markov operators P, P' let $V = \mathsf{P}' - \mathsf{P}$. Then

$$\lim_{h \to 0} \frac{J(\mathsf{P} + hV) - J(\mathsf{P})}{h} = \frac{1}{1-\gamma} \left\langle V q(\mathsf{P}), (\mathsf{Id} - \gamma \mathsf{T}\mathsf{P})^{-*} \nu \right\rangle$$

…we could use first-order methods to minimize J!

[4] Agarwal, Kakade, Lee and Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. JMLR 2021.

[5] We have generalized the definitions of J and q to any Markov operator.

Policy Mirror Descent

(Projected) Gradient Descent?

If we knew to project onto the space of all measurable policies[6] Π...

...we could use projected gradient descent[7] (PGD):

- Start from some $\mathsf{P}_0 \in \Pi$,
- Produce a minimizing sequence iteratively such that, $\forall k \in \mathbb{N}$

$$\mathsf{P}_{k+1} = \mathrm{Proj}_{\Pi}\big(\mathsf{P}_k + \eta \nabla J(\mathsf{P}_k)\big)$$

  with $\eta > 0$ a suitable step-size

Challenge: in practice, it's not clear how to project onto Π.

[6] I will be sloppily confusing P_π with π where it's clear (to me?) from context.
[7] technically, ascent, but we can just minimize $-J$

Mirror Descent

Mirror Descent (MD). Generalizes PGD using a Bregman divergence D instead of a norm:

- For any $k \in \mathbb{N}$ the update step is

$$\mathsf{P}_{k+1} = \underset{\mathsf{P}\in\Pi}{\operatorname{argmin}} \; -\eta \left\langle \nabla J(\mathsf{P}_k), \mathsf{P} \right\rangle + D(\mathsf{P}, \mathsf{P}_k)$$

MD enjoys similar properties to PGD. For example, if J were convex, it would also guarantee convergence[8]!

(Potential) Advantage: carefully choosing D might yield more amenable (e.g. closed-form) solutions for the update step above!

[8] Under some additional assumption on D and Π

"Our" Bregman Divergence

So, let's choose our Bregman divergence.

Generalizing (Xiao2022)[9] from the tabular setting[10], we take

$$D(\mathsf{P}_\pi, \mathsf{P}_{\pi_k}) = \frac{1}{1-\gamma} \int \mathsf{KL}(\pi(\cdot|x), \pi_k(\cdot|x)) \; \rho_{\pi_k}(dx)$$

where:

- $\rho_\pi = (\mathsf{Id} - \gamma \mathsf{T}\mathsf{P}_\pi)^{-*}\nu$ is the "occupancy measure" of π,
- KL is the Kullback-Leibler divergence

[9] Xiao. *On the convergence rates of policy gradient methods.* JMLR 2022.
[10] Tabular setting: $\mathcal{X}$ and $\mathcal{A}$ are sets with finite cardinality.

(Policy) Mirror Descent

Proposition. The (Policy) Mirror Descent step can be performed point-wise, namely the iterative sequence of operators $\mathsf{P}_k = \mathsf{P}_{\pi_k}$ is such that, for any $k \in \mathbb{N}$ and any $x \in \mathcal{X}$

$$\pi_{k+1}(\cdot|x) = \operatorname*{argmin}_{p \in \mathbb{P}[\mathcal{A}]} -\eta \left\langle q_{\pi_k}(\cdot, x), p \right\rangle + \mathsf{KL}(p, \pi_k(\cdot|x))$$

which has closed-form solution

$$\pi_{k+1}(\cdot|x) = \frac{\pi_k(\cdot|x) e^{\eta q_{\pi_k}(x,\cdot)}}{\int_{\mathcal{A}} \pi_k(a|x) e^{\eta q_{\pi_k}(x,a)} \; \pi_k(da|x)}$$

(Policy) Mirror Descent on Finite $\mathcal{A}$

Assumption: let $\mathcal{A}$ have finite cardinality.

Then, by applying the PMD update recursively we have

$$\pi_{k+1}(\cdot|x) = \mathrm{SoftMax}\left(\log \pi_0(\cdot|x) + \eta \sum_{j=0}^{k} q_{\pi_j}(x, \cdot)\right)$$

Where $\mathrm{SoftMax}(q) = \frac{e^q}{\sum_{a \in \mathcal{A}} e^{q(a)}}$.

Note.

While we could generalize the above to generic $\mathcal{A}$, we would be left with an integral at the denominator that we would (likely) be unable to estimate exactly. Studying how such approximation error would propagate will be the subject of future work.

Covergence of PMD

Theorem[11]**.** Let $(\pi_k)_{k\in\mathbb{N}}$ be a sequence generated by PMD with sufficiently large $\eta > 0$. Then,

$$\max_{\pi\in\Pi} J(\pi) - J(\pi_k) \leq O(1/k) \qquad \forall k \in \mathbb{N}$$

Even if J is not convex, PMD converges to the global maximum!

Note: While the objective function J is non-convex, global convergence rates can be proved by gradient domination results. Further, Xiao proved *linear* convergence rates in the tabular case, depending on $\|\frac{d(\mathsf{Id}-\gamma\mathsf{TP})^{-*}\nu}{d\nu}\|_\infty$.

[11] very informal!

Connection to other policy gradient algorithms.

Policy Mirror Descent:

$$\pi_{k+1}(\cdot|x) = \operatorname*{argmin}_{p \in \mathbb{P}[\mathcal{A}]} -\eta \left\langle q_{\pi_k}(\cdot, x), p \right\rangle + \mathsf{KL}(p, \pi_k(\cdot|x))$$

Trust-region Policy Optimizaion (2015):

$$\pi_{k+1}(\cdot|x) = \operatorname*{argmin}_{p \in \mathbb{P}[\mathcal{A}]} - \left\langle q_{\pi_k}(\cdot, x), \frac{p}{\pi_k(\cdot|x)} \right\rangle$$
$$\text{subject to } \mathsf{KL}(\pi_k(\cdot|x), p) \leq \delta$$

IMPLEMENTATION MATTERS IN DEEP POLICY GRADIENTS: A CASE STUDY ON PPO AND TRPO

Logan Engstrom[1*], Andrew Ilyas[1*], Shibani Santurkar[1], Dimitris Tsipras[1], Firdaus Janoos[2], Larry Rudolph[1,2], and Aleksander Mądry[1]

[1]MIT [2]Two Sigma
{engstrom,ailyas,shibani,tsipras,madry}@mit.edu
rudolph@csail.mit.edu, firdaus.janoos@twosigma.com

The 37 Implementation Details of Proximal Policy Optimization

25 Mar 2022 | # proximal-policy-optimization # reproducibility # reinforcement-learning # implementation-details # tutorial

Huang, Shengyi; Dossa, Rousslan Fernand Julien; Raffin, Antonin; Kanervisto, Anssi; Wang, Weixun

Towards a practical Algorithm

Taking Stock

Good news. We have an algorithm to find the best policy, but...

Bad news. For every k we need to know how to evaluate q_{π_k}. But

$$q_{\pi_k} = (\mathsf{Id} - \gamma \mathsf{T} \mathsf{P}_{\pi_k})^{-1} r$$

requires knowledge of the transition operator T!

Challenges:

- In Reinforcement Learning (RL) we **do not** know τ (or T)!
- Even in Dynamic Programming or Optimal Control, where τ is known, it might be too complicated for us to obtain T!

Taking Stock

Good news. We have an algorithm to find the best policy, but...

Bad news. For every k we need to know how to evaluate q_{π_k}. But

$$q_{\pi_k} = (\mathsf{Id} - \gamma \mathsf{T} \mathsf{P}_{\pi_k})^{-1} r$$

requires knowledge of the transition operator T!

Challenges:

- In Reinforcement Learning (RL) we **do not** know τ (or T)!
- Even in Dynamic Programming or Optimal Control, where τ is known, it might be too complicated for us to obtain T!

Idea: let's approximate q_{π_k} with some $\hat{q}_{\pi_k}$ that is more amenable to practical manipulations!

Covergence of "Approximate" PMD

Theorem[12]**.** Let $(\pi_k)_{k\in\mathbb{N}}$ be a sequence generated by the "approximate" PMD step

$$\pi_k(\cdot|x) = \mathrm{SoftMax}\left(\log \pi_0(\cdot|x) + \sum_{j=0}^{k} \hat{q}_{\pi_k}(x,\cdot)\right)$$

where $\hat{q}_{\pi_k}$ are such that $\|\hat{q}_{\hat{\pi}_k} - q_{\hat{\pi}_k}\|_\infty \leq \epsilon_k$ for some $\epsilon_k > 0$. Then,

$$\max_{\pi\in\Pi} J(\pi) - J(\pi_k) \leq O\left(\frac{1+\sum_{j=0}^{k}\epsilon_j}{k}\right) \qquad \forall k \in \mathbb{N}$$

If we can control the ϵ_j (e.g. such that $\epsilon = O(1/j)$), then "approximate" PMD converges to the global maximum!

[12] Again, very informal!

Approximating the action-value function

The need for an approximation $\hat{q}_{\pi_k}$ of the action-value function is shared by most RL algorithms.

Standard approaches minimize the L^2 error between an estimation $\hat{q}_{\pi_k}(x_t, a_t) = \sum_{l \geq 0} \gamma^l r(x_{t+l}, a_{t+l})$ and a parametrized model q_θ

$$\sum_t \left(q_\theta(x_t, a_t) - \hat{q}_{\pi_k}(x_t, a_t)\right)^2$$

In practice one just runs few steps of GD, and the estimators are likely under-optimized. Further, the same model q_θ is used across policies.

We will follow a different approach.

Approximating q_π with World Models

Approximating q_π using World Models

The operator perspective on q_π offers a direct strategy to define a $\hat{q}_\pi$

$$\hat{q}_\pi = (\mathsf{Id} - \gamma \hat{\mathsf{T}} \mathsf{P}_\pi)^{-1} \hat{r}$$

In other words, we need to approximate (or learn!):

- The one-step update of the environment (a "world model" $\hat{\mathsf{T}}$).
- The immediate reward function $\hat{r}$.

Note. We have also to ensure that the definition of $\hat{q}_\pi$ makes sense...

We will rely on standard machine learning tools: kernel methods.

Reward Function. Let $\psi : \Omega \to \mathcal{G}$ be a feature map of a reproducing kernel Hilbert space[13] (rkhs) $\mathcal{G}$. Then, given $n \in \mathbb{N}$ points $(x_i, a_i)_{i=1}^n$,

$$r_n = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} \big(\langle g, \psi(x,a) \rangle_{\mathcal{G}} - r(x,a) \big)^2 + \lambda \|g\|_{\mathcal{G}}^2$$

where $\lambda > 0$ is a regularization parameter.

Notation. We will replace $\hat{r}$ with r_n to highlight the dependency on n.

[13] Namely, $\mathcal{G}$ is a space of functions $g(x,a) = \langle g, \psi(x,a) \rangle$

Learning the Reward Function

Ridge Regression. The quantity r_n admits closed-form solution

$$r_n = S_n^* b = \sum_{i=1}^{n} b_i \psi(x_i, a_i) \qquad \text{where} \qquad b = (K + \lambda \mathsf{Id})^{-1} y$$

where

- $y \in \mathbb{R}^n$ is the vector with entries $y_i = r(x_i, a_i)$,
- $K \in \mathbb{R}^{n \times n}$ the "kernel matrix" with entries

$$K_{ij} = k((x_i, a_i), (x_j, a_j)) = \langle \psi(x_i, a_i), \psi(x_j, a_j) \rangle_{\mathcal{G}}$$

- $S_n : \mathcal{G} \to \mathbb{R}^n$ such that $S_n : g \mapsto (g(x_i, a_i))_{i=1}^n$

Take-home message. We have a "finite" representation of r_n that fits into a machine that we can use in practice!

Can we do the same thing for T? Yes, if...

Remark. Let $\mathcal{G}$ and $\mathcal{F}$ two rkhs over Ω and $\mathcal{X}$ with feature maps $\psi : \Omega \to \mathcal{G}$ and $\varphi : \mathcal{X} \to \mathcal{F}$ respectively.

Then, if the restriction of T to $\mathcal{F}$ takes values in $\mathcal{G}$.

$$(T|_{\mathcal{F}})^* \ \psi(x,a) = \int \varphi(x') \ \tau(dx'|x,a) \qquad \forall (x,a) \in \Omega$$

$\psi(x,a)$ is mapped to the conditional expectation of $\psi(x')$!

Def. T$|_{\mathcal{F}}$ is known as the conditional mean embedding (CME) of τ.

Idea. If we can sample from τ, we can collect a dataset $(x_i, a_i, \varphi(x_i'))_{i=1}^n$ and learn T$|_{\mathcal{F}}$ like we did for r_n.

Conditional Mean Embedding

We formulate the learning problem over Hilbert-Schmidt operators,

$$\tilde{\mathsf{T}}_n = \underset{W \in \mathsf{HS}(\mathcal{F},\mathcal{G})}{\operatorname{argmin}} \; \frac{1}{n}\sum_{i=1}^{n} \|W^*\psi(x_i, a_i) - \varphi(x_i')\|_{\mathcal{F}}^2 + \lambda \big\|W\big\|_{\mathsf{HS}}^2$$

which yields the closed-form solution

$$\tilde{\mathsf{T}}_n = S_n^*(K + \lambda \mathsf{Id})^{-1} Z_n$$

with $Z_n : \mathcal{F} \to \mathbb{R}^n$ such that $Z_n : f \mapsto (f(x_i'))_{i=1}^n$.

Normalization. We then take $\mathsf{T}_n = \frac{\tilde{\mathsf{T}}_n}{\|\tilde{\mathsf{T}}_n\|}$ to ensure $\|\mathsf{T}_n\| = 1$

Take-home message 2. We have a "finite" representation of T_n that fits into a machine that we can use in practice!

Going back to q_π - A Kernel Trick

Can we approximate q_π using r_n and T_n? Yes!

Theorem[14]**.** Let $\mathsf{T}_n = S_n^* B Z_n$ and $r_n = S_n^* b$ for some $B \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, such that $\|\mathsf{T}_n\| \leq 1$. Then

$$q_{\pi,n} = (\mathsf{Id} - \gamma \mathsf{T}_n \mathsf{P}_\pi)^{-1} r_n = S_n^* (\mathsf{Id} - \gamma B M_\pi)^{-1} b = S_n^* b_\pi$$

where $M_\pi = Z_n \mathsf{P}_\pi S_n^* \in \mathbb{R}^{n \times n}$ is the matrix with entries

$$\left(M_\pi\right)_{ij} = \langle \varphi(x_i'), \mathsf{P}_\pi \psi(x_j, a_j) \rangle = \int_{\mathcal{A}} \langle \psi(x_i', a), \psi(x_j, a_j) \rangle \ \pi(da|x_i').$$

So $q_{\pi,n}$ is well-defined AND has a machine-efficient representation!

[14] some assumptions and conditions apply

POWR

POWR Algorithm

We have Policy Mirror Descent with Operator World-models for RL.

- Collect a dataset $(x_i, a_i, x'_i)_{i=1}^n$ of sample transitions to learn T_n (analogously for r_n).
- Choose π_0, for example $\pi_0(\cdot|x)$ uniform for any $x \in \mathcal{X}$.
- **For** $k = 0, \ldots,$
  - Let $q_{\pi_k,n} = (\mathsf{Id} - \gamma \mathsf{T}_n \mathsf{P}_{\pi_k})^{-1} r_n$
  - Let $\pi_{k+1} = \mathrm{SoftMax}\left(\log \pi_0 + \sum_{j=0}^{k} q_{\pi_k,n}\right)$
- **Return:** π_k for any $k \in \mathbb{N}$

Algorithm 1 POWR: POLICY MIRROR DESCENT WITH OPERATOR WORLD-MODELS FOR RL

Input: Dataset $(x_i, a_i, x'_i, r_i)_{i=1}^n$, discount factor $\gamma \in (0,1)$, step size $\eta > 0$, kernel function $k(x,x') = \langle \phi(x), \phi(x') \rangle$ with $\phi : \mathcal{X} \to \mathcal{H}$ as in Proposition 4, initial weights $C_0 = 0 \in \mathbb{R}^{n \times |\mathcal{A}|}$.

/* World Model Learning */
let $E \in \mathbb{R}^{n \times |\mathcal{A}|}$ with rows $E_i = \text{ONEHOT}_{|\mathcal{A}|}(a_i)$.
let $K_\lambda \in \mathbb{R}^{n \times n}$ such that $K_{ij} = k(x_i, x_j)\delta_{a_i = a_j} + n\lambda\delta_{ij}$
let $H \in \mathbb{R}^{n \times n}$ such that $H_{ij} = k(x'_i, x_j)$
compute K_λ^{-1} and $b = K_\lambda^{-1} y$ with $y = (r_1, \dots, r_n) \in \mathbb{R}^n$

/* Policy Mirror Descent */
for $t = 0, 1, \dots, T-1$ **do**:
$\pi_{t+1} = \text{SOFTMAX}(\eta H C_t) \in \mathbb{R}^{n \times |\mathcal{A}|}$
$M_{\pi_{t+1}} = H \odot (\pi_{t+1} E^\top) \in \mathbb{R}^{n \times n}$
$C_{t+1} = C_t + \text{diag}(c) E$ with $c = (\mathsf{Id} - \gamma K_\lambda^{-1} M_{\pi_{t+1}})^{-1} b$
end for

return $\pi_T : \mathcal{X} \to \Delta(\mathcal{A})$ such that $\pi_T(x) = \text{SOFTMAX}(\eta\, H_x C_T)$ with $H_x = (k(x, x_i))_{i=1}^n \in \mathbb{R}^n$.

Convergence

Convergence of POWR - Assumptions

POWR converges under suitable regularity assumptions...

Assumption (Strong Source Condition). There exists [15] $\rho \in \mathcal{P}(\Omega)$ s.t.

$$\|(\mathsf{T}|_{\mathcal{F}})^* C_\rho^{-\beta}\|_{\mathsf{HS}} < +\infty \qquad \text{and} \qquad \|C_\rho^{-\beta} r\|_{\mathcal{G}} < +\infty,$$

for some $\beta > 0$, where $C_\rho = \sum_{a \in \mathcal{A}} \int_{\mathcal{X}} \psi(x,a) \otimes \psi(x,a)\ \rho(dx,a)$.

Notes.

- This is a stronger version of the standard assumption used in supervised learning settings.
- We need it because we will $\mathsf{T}_n \to \mathsf{T}$ and $r_n \to r$ to be in a stronger norm than usual.

[15] We are implicitly asking $r \in \mathrm{range}(C_\rho^\beta)$ and $\mathrm{range}(T|_{\mathcal{F}}) \subseteq \mathrm{range}(C_\rho^\beta)$.

Convergence of POWR

Theorem. Let ρ satisfy the Strong Source Condition. Let the world-model T_n and reward r_n estimators learned from a dataset $(x_i, a_i, x_i')_{i=1}^n$ where (x_i, a_i) are independently sampled from ρ and $x_i' \sim \tau(\cdot|x_i, a_i)$ for $i = 1, \dots, n$.

Then, for any $\delta \in (0,1)$, the iterates produced by POWR converge to the optimal return as

$$\max_{\pi \in \Pi} J(\pi) - J(\pi_k) \leq O\left(\frac{1}{K} + \delta n^{-\frac{\beta}{2+2\beta}}\right)$$

with probability not smaller than $1 - 4e^{-\delta}$

- **Good news:** it converges!
- **Bad news:** maybe not that fast...

Proof Sketch

Proof sketch.

- We know already that PMD with approximate $q_{\pi,n}$ converges with rate $O(1/k+\epsilon)$, if $\|q_{\pi_k,n} - q_\pi\| \leq \varepsilon$ uniformly wrt $k \in \mathbb{N}$.
- The following Lemma gives us an idea of how to control ε:
  Lemma. Assume $\mathsf{T}|_{\mathcal{F}} : \mathcal{F} \to \mathcal{G}$. Then

$$\|q_{\pi,n} - q_\pi\|_\infty \leq O\big(\|r_n - r\|_\infty + \|r\|_\infty \|\mathsf{T}_n - \mathsf{T}|_{\mathcal{F}}\|_{\mathsf{HS}}\big)$$

- Bounding bounding ε boils down to controlling the approximation error of r_n and T_n in $\|\cdot\|_\infty$ norm. This is a supervised setting and we can therefore borrow refined results from the literature[16]

[16] for example Fischer and Steinwart. *Sobolev norm learning rates for regularized least-squares algorithms.* JMLR 2020.

Details

I am hiding a lot of details/questions:

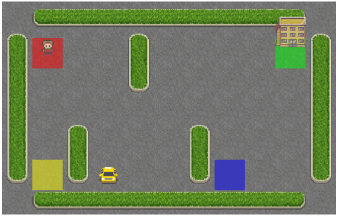
- Constants depending on the key quantities of the problem.
- Minimum sample size n required to make everything work.
- How to choose the step size η?
- How to choose ρ?
- ...

POWR in the “Wild”

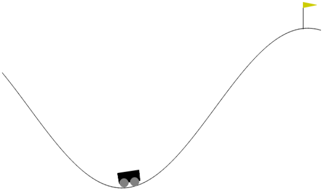
Experiments

For now, we have tried POWR on very small-scale/toy environments.

Frozen Lake-v1

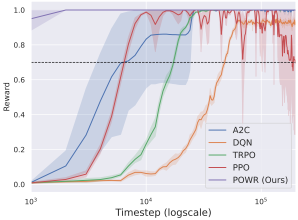


Taxi-v3

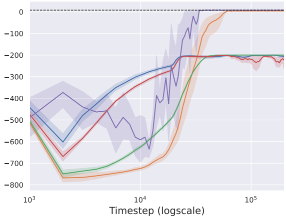


Mountain Car-v0

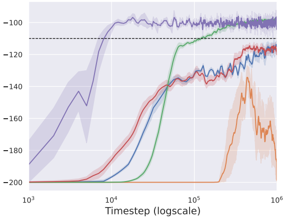
Empirical POWR Sample Efficiency



Frozen Lake-v1



Taxi-v3



Mountain Car-v0

Recap

We...

- Set off to tackle sequential decision making problems...
- Saw how the operator-based perspective offered interesting insights.
- Saw that policy mirror descent converges to the global maximum **provided we can approximate q_π**.
- Proposed an estimator for q_π in terms of a "world model" T_n (and an estimate for the reward r_n).
- Showed that by carefully choosing the spaces where to learn T_n and r_n we can guarantee that POWR:
  - Is well defined
  - Convereges to the global maximum.
- Observed that POWR actually works well in practice.

Open Questions

- (Scaling up) How well does POWR work on more challenging environments?
- (Representation) Are there other choices for $\mathcal{F}, \mathcal{G}$ that guarantee POWR iterates to be $(\mathcal{G}, \mathcal{F})$-compatible? Can we learn them?
- (Efficiency) The usual suspects, Nystrom, Random Features, etc.
- (Infinite Actions) Can we adapt POWR to infinite action spaces?
- (Exploration Vs Exploitation) How to choose the distribution ρ from which we obtain the dataset to train T_n and r_n?

Thanks!

Reinforcement Learning or Dynamic Programming?

Strictly speaking we talk about

- Dynamic Programming (DP) and Optimal Control if τ is known
- Reinforcement Learning (RL) if τ is unknown

Today, we'll be somewhere in between...

Disclaimer

Reinforcement Learning and Dynamic Programming have a relatively long history, dating back to the late 1950s from the work of Bellman[17], Samuel[18] and Howard[19].

They are closely related with Optimal Control and both very active fields, with a plethora of approaches and techniques developed over the years.

However, I am going to blatantly ignore all of that and give a very biased and focused talk on a specific and relatively novel perspective on how to tackle them.

[17]Bellman, R.*Dynamic Programming*. Princeton University Press. 1957
[18]Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*. 1959
[19]Howard, R. A. *Dynamic Probabilistic Systems*. Wiley. 1960.

References

However, here is a list of references to get started on the topic:

- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.
- Hotchkiss, G. N., & Mason, S. A. (2019). *Algorithmic Dynamic Programming*. Springer.
- Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control* (Vol. 1). Athena Scientific.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
- Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Morgan & Claypool.
- Lapan, M. (2020). *Deep Reinforcement Learning Hands-On* (2nd ed.). Packt Publishing.

Going back to q_π

Can we approximate q_π using r_n and T_n? Yes, if π is $(\mathcal{G}, \mathcal{F})$-compatible!

Def. A policy π is $(\mathcal{G}, \mathcal{F})$-compatible if $(\mathsf{P}_\pi)|_{\mathcal{G}}$ has range in $\mathcal{F}$.

Theorem. Let $\mathsf{T}_n = S_n^* B Z_n$ and $r_n = S_n^* b$ for some $B \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, such that $\|\mathsf{T}_n\| \leq 1$. For any $(\mathcal{G}, \mathcal{F})$-compatible policy π,

$$q_{\pi,n} = (\mathsf{Id} - \gamma \mathsf{T}_n \mathsf{P}_\pi)^{-1} r_n = S_n^* (\mathsf{Id} - \gamma B M_\pi)^{-1} b$$

where $M_\pi = Z_n \mathsf{P}_\pi S_n^* \in \mathbb{R}^{n \times n}$ is the matrix with entries

$$\left(M_\pi\right)_{ij} = \langle \varphi(x_i'), \mathsf{P}_\pi \psi(x_j, a_j) \rangle = \int_{\mathcal{A}} \langle \psi(x_i', a), \psi(x_j, a_j) \rangle \ \pi(da|x_i').$$

So $q_{\pi,n}$ is well-defined AND has a machine-efficient representation!

Does POWR "Work"?

Two main questions:

- Are POWR's iterates $(\mathcal{G}, \mathcal{F})$-compatible? And why do we care?
- When (if ever) does POWR converge?

Guarnteeing Compatibility - Factoring $\mathcal{F}$ and $\mathcal{G}$

We restrict to the following choice for $\mathcal{F}$ and $\mathcal{G}$:

- $\mathcal{H}$ be a rkhs with feature map $\phi : \mathcal{X} \to \mathcal{H}$.
- $\mathcal{F} = \mathcal{H} \otimes \mathcal{H}$ with $\varphi(x) = \phi(x) \otimes \phi(x)$,
- $\mathcal{G} = \mathbb{R}^{|\mathcal{A}|} \otimes \mathcal{H}$ with[20] $\psi(x, a) = e_a \otimes \phi(x)$.

Then – recalling that $\mathcal{A}$ is a finite set – we have the following,

Proposition. A policy π is $(\mathcal{G}, \mathcal{F})$-compatible if and only if there exist $p_a \in \mathcal{H}$ such that $\pi(a|\cdot) = \langle p_a, \phi(\cdot) \rangle_{\mathcal{H}}$ for and $a \in \mathcal{A}$.

It is enough to check that all $\pi(a|\cdot)$ "belong" to $\mathcal{H}$ to guarantee $(\mathcal{F}, \mathcal{G})$-compatibility!

[20] Here, e_a is the a-th element of the canonical basis of $\mathbb{R}^{|\mathcal{A}|}$ (assume an order on $\mathcal{A}$).

Sobolev Spaces to the Rescue!

Theorem. Let $\mathcal{X} \subset \mathbb{R}^d$ be compact, $\mathcal{H} = W^{2,s}(\mathcal{X})$ the Sobolev space with smoothness $s > d/2$. Let $\pi_0(a|\cdot) \propto e^{\eta q_0(\cdot,a)}$ for some $q_0(\cdot,a) \in \mathcal{H}$ for all $a \in \mathcal{A}$.

$\implies$ all iterates produced by POWR are $(\mathcal{F}, \mathcal{G})$-compatible.

Proof sketch. The key is to show recursively that

- If π_k is $(\mathcal{G}, \mathcal{F})$-compatible, then the approximate $q_{\pi_k,n}$ belong to $\mathcal{H}$ and,
- The $\mathrm{SoftMax}$ operator applied to previous $q_{\pi_k,n}$ yields a $(\mathcal{G}, \mathcal{F})$-compatible policy π_{k+1}

Algorithm 1 POWR: Policy mirror descent with Operator World-models for Rl

Input: Dataset $(x_i, a_i, x'_i, r_i)_{i=1}^n$, discount factor $\gamma \in (0,1)$, step size $\eta > 0$, kernel function $k(x, x') = \langle \phi(x), \phi(x') \rangle$ with $\phi : \mathcal{X} \to \mathcal{H}$ as in Proposition 4, initial weights $C_0 = 0 \in \mathbb{R}^{n \times |\mathcal{A}|}$.

```
/* World Model Learning */
```
let $E \in \mathbb{R}^{n \times |\mathcal{A}|}$ with rows $E_i = \text{ONEHOT}_{|\mathcal{A}|}(a_i)$.
let $K_\lambda \in \mathbb{R}^{n \times n}$ such that $K_{ij} = k(x_i, x_j)\delta_{a_i = a_j} + n\lambda\delta_{ij}$
let $H \in \mathbb{R}^{n \times n}$ such that $H_{ij} = k(x'_i, x_j)$
compute K_λ^{-1} and $b = K_\lambda^{-1} y$ with $y = (r_1, \dots, r_n) \in \mathbb{R}^n$

```
/* Policy Mirror Descent */
```
for $t = 0, 1, \dots, T-1$ **do:**

$\pi_{t+1} = \text{SOFTMAX}(\eta H C_t) \in \mathbb{R}^{n \times |\mathcal{A}|}$
$M_{\pi_{t+1}} = H \odot (\pi_{t+1} E^\top) \in \mathbb{R}^{n \times n}$
$C_{t+1} = C_t + \text{diag}(c) E$ with $c = (\mathsf{Id} - \gamma K_\lambda^{-1} M_{\pi_{t+1}})^{-1} b$

end for

return $\pi_T : \mathcal{X} \to \Delta(\mathcal{A})$ such that $\pi_T(x) = \text{SOFTMAX}(\eta\, H_x C_T)$ with $H_x = (k(x, x_i))_{i=1}^n \in \mathbb{R}^n$.

Does POWR "Work"?

Two main questions:

- Are POWR's iterates $(\mathcal{G}, \mathcal{F})$-compatible? Yes!
- When (if ever) does POWR converge?