Notes on
Theoretical Foundations of Reinforcement Learning, FOCS 2020
Kumar Krishna Agrawal [1]

[1]This is a working draft. Any errors in the text are due to the author.

Contents

1 Introduction

The primary goal is to minimize regret to a policu class

$$\max_{\pi \in \Pi} \sum_{x,r} r_{\pi(x)} - r(a)$$

Is there a reasonable lower bound to the complexity of the learning problem?
COnsider the exponentially large tree, one rewarding leaf such that the algorithm must try all leaves to find reward. This gives is $\mathcal{A}^H$ sample complexity. Here the exponential case is slightly different, in the sense that per-episode we only get to improve the policy that took the action.
Consequence: Reinforcement Learning is a family of problems with core set of challenges but varying assumptions/settings.
Ex: Consider the two different horizons

- success after H steps
- success γ^t discounted t steps into the future. (interpretation of discount γ: termination with probability $1-\gamma$ and no discount giving us an approximate $H \sim \frac{1}{1-\gamma}$

1.1 Challenges

- **Credit Assignment**
- **Exploration**

  How do you collect the right data you need to learn?
- **Generalization**

  How to generalize from past examples?
- **Exploration + Generalization = Contextual Bandits**

  Repeatedly : See features x, Choose actions a, See reward r

  Minimize regret over IID sequence of (x, r)

  $$\max_{\pi \in \Pi} \sum_{x,r} r_{\pi(x)} - r_a$$

  - Thompson Sampling (Tho33) : First "bandit" style algorithms
  - EXP4 (ACBFS02): Considers the adversarial settings.
  - Epoch Greedy (LZ07) : Polytime given an oracle
  - Deployment (LCLS10) : Web recommendation application
- **Credit Assignment + Exploration = MDP Learning**
  - Observe state s, action a, next state s' transitions
  - Build an imperfect model of the world $\hat{T}(s'|s,a), \hat{R}(r|s,a)$
  - Plan with imperfect model to reach unobserved transitions and maximize rewards.
  - Result : Poly(S, A) time complexity

1.2 Tabular MDP Learning

- E^3 (KS02), Rmax (BT02) : Polynomial time learning is possible
- Delayed-Q : (SLW+06), linear in states is possible
- UCRL : (JOA10) near-optimal regret is possible
- UCB-B : (JAZBJ18) model-free near-optimal regret

2 Information Theory

Consider finite-horizon episodic MDP, with horizon H.

- $x_1 \sim P_1$
- for h = 1, ..., H
- observe $x_h \in \mathcal{X}$
- take action $a_h in \mathcal{A}$
- observe reward $r_h \in [0, 1]$
- transition to $x_{h+1} \sim T(\cdot|x_h, a_h)$

Goal: Find the policy $\pi : \mathcal{X} \to \mathcal{A}$ maximizing the value of a policy $V(\pi) = \mathbb{E}_\pi[\sum_{h=1}^{H} r_h]$
PAC Learning : Output $\hat{\pi}$ with $V(\hat{\pi}) \geq \max_\pi V(\pi) - \epsilon$. Minimize number of episodes required.
Denote by V^*, π^* the optimal value function and the optimal policy.
Regret: Minimize $T \cdot V^*$ - (Learner Reward)

2.1 Solution Concepts

With large state-space, our search space $\mathcal{X} \to \mathcal{A}$ is huge. To generalize, we restrict the space in which we search for solutions by one of the following:

- Policy Search : Restricting the set of possible solution candidates to $\Pi \subset \{\mathcal{X} \to \mathcal{A}\}$.

  multiclass linear classifiers : $x \mapsto \arg\max_a \theta_a^T x$

- Value-based: Approximate action-value $Q \in \mathcal{Q} \subset \{\mathcal{X} \times \mathcal{A} \to \mathbb{R}\}$

  every Q-function encodes a policy, $\pi_Q : x \mapsto \arg\max_a Q(x, a)$

$$
\begin{aligned}
Q_h^*(x, a) &:= \mathbb{E}\Big[\sum_{h'=h}^{H} r_{h'} | x_h = x, a_h = a, a_{h+1:H} \sim \pi^*\Big] \\
&:= \underbrace{\mathbb{E}\Big[r_h + \max_{a'} Q'_{h+1}(x', a') | x_h = x, a_h = a\Big]}_{\text{Bellman's Update}}
\end{aligned}
$$

- Model based: Approximate model M in $\mathcal{M}$.

2.2 Statistical goal

We want to design algorithms which recover $\hat{\pi}$, where $V(\hat{\pi}) \geq \max_{\pi \in \Pi} V(\pi) - \epsilon$ while minimizing number of episodes required. Here Π is the restricted class of induced policies.

Goal : $poly(|\mathcal{A}|, H, comp(\mathcal{F}), 1/\epsilon)$, where

- $comp(\mathcal{F})$: statistical complexity of policy class (as in supervised learning)

  $comp(\mathcal{F})$: log $|\mathcal{F}|$ if finite classes

  $comp(\mathcal{F})$: d for d-dimensional linear predictors

- $\frac{1}{\epsilon}$ is the accuracy parameter.

Note : The bounds are independent of $|\mathcal{X}| \implies$ suggests generalization across states

2.3 RL is not like supervised learning

Consider the problem : repeat n samples $(s, a) \sim \mathcal{D}$, which reveals $y = Q^*(s, a)$ (as in contextual bandits) . Given a function class $\mathcal{F}$, we consider searching for the best approximation of the *optimal* Q-function under MSE loss.

$$\text{solve}\hat{Q} = \arg\min_{Q \in \mathcal{F}} \sum_{s,a} (Q(s_i, a_i) - y_i)^2$$

From standard supervised learning bounds, we know

$$\mathbb{E}_{\mathcal{D}}\left[(\hat{Q}(s, a) - Q^*(s, a))^2\right] \leq \frac{comp(\mathcal{F})}{n}$$

What about policy performance?
Particularly, the greedy encoded policy $\hat{\pi} : s \mapsto \arg\max_a \hat{Q}(s, a)$. We cannot comment much on the quality of the policy without additional knowledege. Particularly, if $\exists \hat{\pi}(x) \in \mathcal{A}$, not supported by $\mathcal{D} \implies$ we might have $\hat{Q}(x, \hat{\pi}(x)) >> Q^*(x, \hat{\pi}(x))$

2.4 "Distribution Shift"

States where we evaluate $\hat{\pi}$ would be different from the distribution we train the Q function. This requires out of distribution generalization. Conceptual ways to address this

- Assumptions about environment : not many data-distributions available. so we collect data from all these candidates, we shouldn't have support issues.
- Assumption about $\mathcal{F}$: OOD generalization by "extrapolation"
- combination of both.

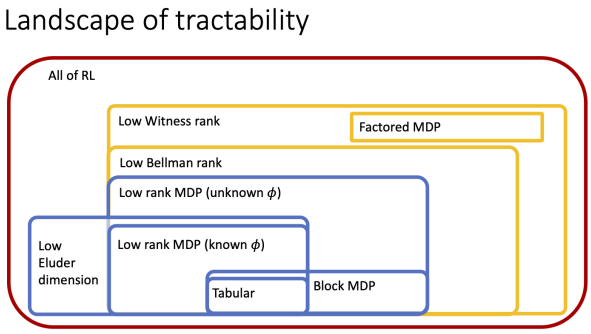

Figure 1: The landscape of tracatability for reinforcement learning algorithms.(AKL20)

2.4.1 Contextual Bandits

Unit time horizon, with finite actions $\mathcal{A}$, reward function $\in \mathcal{F}$ such that $f^*(x,a) = \mathbb{E}[r|x,a]$ Collect n samples, (x,a,r) with $a \sim Unif[\mathcal{A}]$. We try to recover the reward function by solving the least squares problem

$$\hat{f} = \arg\min_{f \in \mathcal{F}} (f(x_i, a_i) - r_i)^2$$

define $\hat{\pi}(x) = \arg\max_a \hat{f}(x,a)$

Proposition 1. $V(\pi^*) - V(\hat{\pi}) \leq \sqrt{\frac{comp(\mathcal{F})}{n} \cdot |\mathcal{A}|}$.

Notes : Here $|\mathcal{A}|$ is the effective number of distributions in the problem. Each policy induces different state-action distributions, but all these distributions are covered upto multiplicative factor $|\mathcal{A}|$ by a uniform distribution over actions. In the multi-step reinforcement learning, this implies $|\mathcal{A}|^H$ which is exponential in H (not good enough).

2.4.2 Linear Bandits

Assume time horizon is 1, fixed starting state, well-specified linear reward $< \theta^*, \phi(a) >= \mathbb{E}[r|a]$ Solution : Collect n samples on an approximate basis (related : (AK08; HKM14))

Proposition 2. $V(\pi^*) - V(\hat{\pi}) \leq \sqrt{\frac{poly(d)}{n}}$, *where* $comp(\mathcal{F}) = d$ *and independent of* $\mathcal{A}$.

Intuition : Linear function class allows extrapolating predictions from basis to entire space.

2.5 The Landscape of tractability in RL

2.5.1 RL with Linear functions

Value based RL where $\mathcal{F}$ is linear functions of known feature map $\phi(x,a)$. We note that well specified linear function class direcly support extrapolation.

Question 1. *Is* $Q^* \in \mathcal{F}$ *sufficient for multi-step reinforcement learning?*

- Evidence no: insufficient in batch setting(WFK20), insufficient with many actions(WAS20)

- lesson : error amplification might be troubleson, linear extrapolation isn't the only issue.

Low rank MDP (or linear MDP)

- Rewards and transitions are linear in features. $T(x'|x,a) = \phi(x,a)\mu(x')$
- already implies that $Q^* \in \mathcal{F}$ and enables credit assignment.

Algorithm 1 LSVI-UCB

Optimistic dynamic programming

$$\hat{\theta}_h = \arg\min_{\theta} \sum \Big(\underbrace{\langle \theta, \phi(x_h, a_h) \rangle}_{\text{linear predictor}} - \underbrace{r_h - \max_{a'} \hat{Q}_{h+1}(x_{h+1}, a')}_{\text{regression target}} \Big)^2$$

Define $\hat{Q}_h(x,a) = \langle \hat{\theta}, \phi(x,a) \rangle + \text{bonus}_h(x,a)$
Collect data with greedy policy with $Q_1, ..., Q_H$

Theorem 1. *(JYWJ20)* $\tilde{O}(\sqrt{d^3H^3T})$ *regret in* T *rounds in the low-rank MDP*

LSVI-UCB Analysis

- Optimistic regret decomposition : If $\hat{Q} \geq Q^*$ pointwise (i.e optimistic) then

$$\underbrace{Q_h^*(x, \pi^*(x)) - Q_h^*(x, \hat{\pi}(x))}_{\text{true value - optimal lookahead}} \leq \underbrace{\hat{Q}_h(x, \hat{\pi}(x)) - Q_h^*(x, \hat{\pi}(x))}_{\text{depends only on } \hat{\pi}}$$
$$\leq \mathbf{b}_h(x, \hat{\pi}(x)) + \mathbf{e}_h(x, \hat{\pi}(x)) + \mathbb{E}[\hat{Q}_{h+1}(x', \hat{\pi}(x')) - Q_{h+1}^*(x', \pi^*(x'))|x, \hat{\pi}(x)]$$

- if $e_h \leq b_h$ pointwise then

$$\textit{Regret} \leq 2 \cdot \sum_h \text{bonus}_h(x_h, \hat{\pi}(x_h))$$

- Ensuring optimism : standard well-specified linear least-squares analysis (almost)
- Potential function : bonuses cannot be large forever

$$\text{bonus}^2 = \phi^T \Sigma^{-1} \phi, \Sigma \leftarrow \Sigma + \phi\phi^T$$

Open Problem : Computationally efficient approach to get optimal rate in this setting?

2.5.2 Eluder dimension

Combinatorial parameter that captures worst-case extrapolation

Definition 1 (Eluder dimension): *Point z in ϵ-dependent on $z_1, ..., z_n$ if predictions on $z_1, ..., z_n$ constrain prediciont on z:*

$$\forall f, f' : \sqrt{\sum_{i=1}^{n}(f(z_i) - f'(z_i))^2} \leq \epsilon \implies |f(z) - f'(z)| \leq$$

Eluder dimension $dim(\mathcal{F}, \epsilon)$ is the length of the longest independent sequence.

- Potential function : If bonus is large the point must be dependent on few disjoint subsequences.
- Can give $\sqrt{T}$ regret rates for bandits (RVR13)
- RL requires more assumptions (WSY20)

Discussion

- When is the eluder dimension small?

  Linear function : O(d)

  Generalized linear functions : $f(x) \mapsto \sigma(\langle\theta, x\rangle)$, smooth σ : O(d)

  ReLU : $f(x) \mapsto max(0, \langle\theta, x\rangle)$: $exp(d)$ (Li, Kamath)

Open Problem : What non-linear settings admit small Eluder dimension?

2.5.3 Bellman Rank in MDPs

In the "low-rank" setting, we assume that dynamics $T(s'|s, a)$ admits a low-rank factorization in terms of functions ϕ, μ. In such settings, for any π and any $g : \mathcal{X} \mapsto \mathbb{R}$

$$\begin{aligned}\mathbb{E}_\pi g(x_h) &= \mathbb{E}_\pi \int T(x_h|x_{h-1}, \pi(x_{h-1}) g(x_h) d(x_h) \\ &= \langle \mathbb{E}_\pi \phi(x_{h-1}, \pi(x_{h-1})), \int \mu(x_h) g(x_h) d(x_h) \rangle\end{aligned}$$

All expectations live in a d-dimensional space $\implies$ "a few" distributions

Definition 2 (Average Bellman error): *Expected value of residual between f's prediction at $h,$, $h+1$ on data distribution of π.*

$$\mathcal{E}_h(\pi, f) := \mathbb{E}\Big[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1}) | x_h \sim_p i, a_h \sim \pi_f\Big]$$

where π_f is the greedy policy encoded by value function f.

In low rank MDP $\mathcal{E}_h(\pi, f) = \langle \alpha(\pi), \beta(f) \rangle$

Definition 3 (Bellman rank): $\max_h rank(\mathcal{E}_h)$ *where* $\mathcal{E}_h$ *is a matrix using* $\pi \in \Pi, f \in \mathcal{F}$

Sufficiency of Bellman Rank

- Assune that $Q^* \in \mathcal{F}$ (realizability), Bellman rank $\leq M$.

  Theorem 2. *(JKA+17) OLIVE learns an* ε *sub-optimal policy with sample complexity*

  $$\tilde{O}(M^2|\mathcal{A}|H^3 comp(\mathcal{F})/\epsilon^2)$$

- with alternative definition, we can eliminate dependence on $|\mathcal{A}|$
- $\sqrt{T}$-regret rate is achievable (DPWZ19)

Algorithm 2 OLIVE

Optimistic V^* : Pick surviving $\hat{f} \in \mathcal{F}$ to maximize $\mathbb{E}\Big[f(x_1, \pi_f(x_1))\Big] = V^f(\pi_f)$
Actual Value : Collect trajectories with $\pi_{\hat{f}}$, estimate $V(\pi_{\hat{f}})$
Check guess : Output $\pi_{\hat{f}}$ if $V(\pi_{\hat{f}}) \sim V^{\hat{f}}(\pi_{\hat{f}})$
Eliminate all f for which $\mathcal{E}_h(\pi_{\hat{f}}, f) \neq 0$ at some h

---

- First observation : Each iteration requires $poly(|\mathcal{A}|, H, comp(\mathcal{F})/\varepsilon)$ samples

  Key issue : How many iterations?

OLIVE Analysis

Recall $\mathcal{E}_h(\pi, f) := \mathbb{E}\Big[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1})|x_h \sim_p i, a_h \sim \pi_f\Big]$

- Claim 1: Q^* is never eliminated, since $\mathcal{E}(\pi, Q^*) = 0$

  Q^* satisfies Bellman optimality equation : $Q^*(x, a) = \mathbb{E}\Big[r + Q^*(x', \pi(x'))|x, a\Big]$

  Claim 1 + Optimism $\implies$ near-optimality upon termation

- Claim 2: Telescoping performance decomposition $V^f(\pi_f) - V(\pi_f) = \sum_{h=1}^{H} \mathcal{E}_h(\pi_f, f)$

  $\implies$ $\hat{f}$ eliminated

- Claim 3 : Iterations $\leq MH$

Examples

- block MDP

  Discrete hidden state space

  Observations from emission distributions

  Roll-in policy induces distribution over hidden state

  $$\mathcal{E}_h(\pi, f) := \sum_s P^\pi(s)\Big[f(x, a) - r - f(x', \pi_f(x'))|x \sim s, a \sim \pi_f\Big]$$

  Bellman rank = # of hidden states, for any function class $\mathcal{F}$

- low rank MDP

  $\mathcal{E}_h(\pi, f) = \langle \mathbb{E}_\pi \phi(x_{h-1}, \pi(x_{h-1})), \int \mu(x_h) g(x_h) d(x_h) \rangle$

  Bellman rank = rank of transiiton operator, for any function class $\mathcal{F}$

  $\implies$ low rank MDP with "feature selection" or unknown ϕ is statistically tractable

- Linear Completeness

  Assumption : Bellman backup of linear function is linear in current features

  For any $\theta, \exists w$ such that

$$\underbrace{(\mathcal{T}\theta)(x,a)}_{\text{Bellman backup operator}} := \mathbb{E}[r + \max_{a'} \langle \theta, \phi(x', a') \rangle | x, a] = \langle w, \phi(x, a) \rangle$$

  Standard assumption in analysis of dynamic programming algorithms

$$\begin{aligned}
\mathcal{E}_h(\pi, \theta) &= \mathbb{E}_\pi \Big[\langle \theta, \phi(x,a) \rangle - r - \max_{a'} \langle \theta, \phi(x', a') \rangle \Big] \\
&= \mathbb{E}_\pi \Big[\langle \theta - w, \phi(x,a) \rangle \Big] \\
&= \langle \theta - w, \mathbb{E}_\pi \langle phi(x,a) \rangle \rangle \qquad \text{(Uses alternative definition)}
\end{aligned}$$

  Exploits structure of function class $\mathcal{F}$, not environment dynamics

  More like extrapolation argument for linear classes (ZLKB20)

Summary : Bellman rank captures structure of environment and $\mathcal{F}$
Summary of value-based methods

- Linear methods

  Positive results require more than realizability, e.g completeness conditions

  Current thinking is that realizability alone insufficient

- Nonlinear methods require some dynamics assumptions for low Bellman rank

- Two conceptual approaches for handling distribution shift

  Do neural networks (or other nonlinear classes) enable extrapolation?

  Optimal rates for extrapolation approaches?

2.5.4 Factored MDP

- State is a d-dimensional discrete vector. Transition operator factorizes

$$T(x'|x,a) = \Pi_{i=1}^{d} T_i(x'[i] | x[pa(i)], a)$$

- Binary state vars : unfactored $2^{2d}|\mathcal{A}|$ parameters, factored $2^{L+1}d|\mathcal{A}|$ parameters, L $= \max_i |pa(i)|$
- Statistics : $\tilde{O}(\sqrt{\#\{params\} \cdot T})$ regret with known parent structure. Can also learn structure.

- Computation

  Planning with a known model is computationally hard (what combinatorial models are computationally tractable?)

  Optimal policy cannot be represented by a poly-sized circuit

- Statistical separation:

  (SJK+19) For value-based methods, realizability alone is insufficient

A complexity measure

- For model based RL, assume calss $\mathcal{M}$ of candidate duynamics model, and $T \in \mathcal{M}$
- Witness rank $= \max_h \text{rank}(\mathcal{W}_h)$

$$\mathcal{W}_h(\pi, M) = \mathbb{E}[||M(\cdot|x_h, a_h) - T(\cdot|x_h, a_h)||_{TV}|x_h \sim \pi, a_h \sim \pi_M]$$

- Witness rank $\leq$ Bellman rank (for $\mathcal{F} = plan(\mathcal{M})$)

  Theorem 3. *(SJK+ 19) Sample complexity poly(witness rank, $|\mathcal{A}|, H$, comp($\mathcal{M}$)) achievable*

- Factored MDP : witness rank $\leq \#params$

Model-based needs strong realizability conditions, but can succeed where valu-based fails.

3 Optimization

3.1 RL as optimization

Consider the discounted MDP $(\mathcal{X}, \mathcal{A}, R, T, \gamma)$, where this is infinite-horizon MDP. Since we care about near-term actions more, $\gamma \in (0, 1)$, we get that effective horizon $H = \frac{1}{1-\gamma}$. Setting up the optimization objective, we consider the PAC goal for some policy π_θ

$$\max_\theta V(\pi_\theta) = \mathbb{E}^{\pi_\theta}\Big[\sum_t^\infty \gamma^t r(x_t, a_t)\Big]$$

We want π_θ which gives large-expected discounted reward. We consider key questions:

- What is the optimization landscape?
- Role of eplorationa and parameterization?
- Convergence and sample complexity of (stochastic) gradient methods?

3.1.1 Stochastic policies

We consider policies $\pi_\theta : \mathcal{X} \mapsto \Delta(\mathcal{A})$, which are usually smooth in the parameters and hence amenable to gradient based optimization. For example, consider softmax form : $\pi_\theta(a|x) \propto e^{f_\theta(x,a)}$ where f_θ is some function of the state, action. Based on the choice of f we get

- tabular MDP : with small $\mathcal{A}, \mathcal{S}$, we consider $f_\theta(x,a) : \theta_{x,a}$ which is single parameter per (x, a). This is fully expressible class of policies (i.e any policy can be represented using this parametrization).
- Linear : $f_\theta(x,a) = \theta^T \phi(x,a)$
- Neural : $f_\theta(x,a) = NN(x,a;\theta)$
- Gaussian : $f_\theta(x,a) = ||\phi(x,a) - g_\theta(x,a)||^2$ (often for control settings)

3.1.2 State distributions and value functions

Define the state distribution induced by a policy π to get

$$d^\pi_{x_0}(x) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t Pr^\pi(x_t = x|x_0)$$

where x_0 is some initial state. We use notation d^π_μ when $x_0 \sim \mu$. Similarly we define $d^\pi_{x_0,a_0}(x,a)$ with $d^\pi_{x_0}(x,a)$ when $a_0 \sim \pi(\cdot|a_0)$
The value function of π is the cumulative reward when actions are drawn according to π:

$$V^\pi(x_0) = \frac{1}{1-\gamma}\mathbb{E}_{x,a\sim d^\pi_{x_0}}[r(x,a)]$$
$$Q^\pi(x,a) = \mathbb{E}[r(x,a) + \gamma V^\pi(x')|x,a]$$

3.2 Policy gradient methods

We have the optimization objective for RL as $\max_\theta V^{\pi_\theta}(x_0)$. We want to use $\nabla_\theta V^{(t)}(x_0)$ for first-order updates on value of policy as

$$\theta_{t+1} = \theta_t + \eta \nabla_\theta V^{(t)}(x_0) \qquad \text{(where } V^t = V^\pi|\pi = \pi_{\theta_t}\text{)}$$

Theorem 4. *((Wil92; SMSM99)) The gradient of value function w.r.t policy parameters*

$$\nabla_\theta V^{\pi_\theta}(x_0) = \mathbb{E}_{(x,a)\sim d^{\pi_\theta}_{x_0}}\Big[\underbrace{\nabla_\theta \log \pi_\theta(a|x)}_{\text{closed form}} \quad \underbrace{Q^{\pi_\theta}(x,a)}_{\text{unbiased estimator } \hat{Q}}\Big]$$

Here $\hat{Q}(x_i,a_i) = \sum_{t=i}^{\infty} r(s_i,a_i)$ is unbiased and estimated using trajectories from π_θ. This estimate can be used to run stochastic gradient ascent (REINFORCE, (Wil92)).

3.2.1 History

- First algorithm: REINFORCE(Wil92), actor-critic (KT00), function approx (SMSM99)
- Empirical progress : improvement from optimization (i) trust region (SLA+15; SWD+17) (ii) variance reduction (iii) entropy regularized SAC (HZAL18)
- Theoretical basis : convergence to stationary points under smoothness (SMSM99; RM51)
- policy improvement : we want a policy $\pi \in \Pi$ which is close/better than π_0 without constraints like global optimality (KL02)
- global optimality : Conversative policy iteration (CPI) (KL02) describes an algorithm which under certain conditions of (i) exploration (ii) parameterization, we get global optima.

3.2.2 Optimization Landscape

We know there exists MDP such that $V^{\pi_\theta}(x_0)$ is non-concave in θ (i.e we're not guaranteed unique global optimum.).

Theorem 5. *(AKLM19) There is an MDP where O(H)-order gradients have norm at most e^{-H} and where the corresponding policy gets e^{-H} expected reward.*

The result suggests that for such MDPs, there are policies which reach the final state with exponentially small probability. If we look at the gradients, they are small (irrespective of the order). This implies, that any gradient method which minimized first/higher order derivative, it will not necessarily find good policies. This is not a statistical problem, since the exact gradients here are extremely small and we cannot fix this with data.

3.2.3 Optimization challenges in RL

- Supervised learning : gradient descent "generally works?" in practice, not sensitive to initialization. saddle points not necessarily a problem
- Reinforcement learning : Many RL problems have "very flat" regions. Small gradients can necessarily be due to poor exploration.

How to fix?

- Favorable state distributions (given, or can we construct?)
- Favorable reward structures : bring the reward closer to the agent (imitaion learning, reward shaping?)

3.2.4 The initial distribution

The objective is to understand what constitutes favorable state distributions, how to construct them. Optimization problem $\max_\theta V^{\pi_\theta}(x_0)$. For algorithm, se assume state $x_0 \sim \mu$, e.g. uniform over states in chain. This allows us to write (i) gradients (ii) expectations, state distributions of policy as a function of μ

- COnvergence and asample complexity of (stochastic) gradient methods?
- Role of exploration and parameterization?
- How to construct favorable initial distributions algorithmically?

3.2.5 Convergence properties in tabular setting

Primarily to understand in a simpler setting. Here we consider $\pi_\theta(a|x) \propto e^{\theta_{x,a}}$, with the vanilla policy gradient algorithm : $\theta_{t+1} = \theta_t + \eta \nabla_\theta V^{(t)}(\mu)$.

Theorem 6. *(AKLM19) Suppose $\mu(x) \geq 0 \forall x \in \mathcal{X}$ and stepsize $\eta \leq \frac{(1-\gamma)^3}{8}$, for all x, $V^{(t)}(x) \to V^*$ as $t \to \infty$ (under stronger assumptions we can give rates (MJTS20))*

Intuition : We start with some non-zero distribution over all states, actions (x, a). So we try out everything, and hopefully find something globally optimum. We are converging to a dertrministic policy, which means several action probabilities become 0. Sice several entries being 0, we get a competition between optimization and exploration. As we get closer to an optimum, the exploration becomes worse and worse. Conjecture : Convergence is exponentially slow $\mathcal{S}, \mathcal{A}$in the worst case. Intuitively, we are in this ill-conditioned case, parameters near optimal solution where the landscape is flat, gradient is very small.

3.2.6 Natural Policy Gradient

Introduce preconditioning for improving the optimization landscape. (use a preconditioning matrix) (Kak01) introduced Natural Policy Gradient. Uses a Fisher information based preconditioner. Inspiration for practical approaches like TRPO/PPO. Simple form for tabular softmax parameterization

$$\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} Q^{(t)}, \qquad \text{and} \qquad \pi_{t+1}(a|x) \propto \pi_t(a|x) e^{\eta Q^{(t)}}$$

This looks like multiplicative weights, which has good theory in TCS. (we have non-concave maximization objective)

Theorem 7. *(AKLM19) The initial state distribution doesn't have to be exploratory in the tabular setting. Using $\mu = \delta_{x_0}, \theta_0 = 0$, setting $\eta = (1-\gamma)^2 log|\mathcal{A}|$, for all t we have*

$$V^*(x_0) - V^{(t)}(x_0) \leq \frac{2}{(1-\gamma)^2 t}$$

- dimension free convergence, no dependence on $|\mathcal{X}|, |\mathcal{A}|$
- no reliance on state coverage under μ
- some related work : (EDKM09; NJG17; GSP19; SEM20; AYBB+19)
- assumes exact gradient (no sampling), or know Q
- still require exploratory μ for sample complexity (if we have samples, the noise does interact with how exploratory μ)

Proof ideas

- Performance difference lemma (KL02)

$$V^{\pi}(x_0) - V^{\pi'}(x_0) = \frac{1}{1-\gamma}\mathbb{E}_{x\sim d^{\pi}_{x_0}}\mathbb{E}_{a\sim\pi(\cdot|x)}\Big[Q^{\pi'}(x,a) - V^{\pi'}(x)\Big]$$

  Consider $\pi = \pi^*$, and $\pi' = \pi^{(t)}$, we get something like regret. The multiplicative updates are akin to fixing $\pi*$, and updating $\pi^{(t+1)}$

- Linearize regret using above lemma, instead of concavity
- Yields $\frac{1}{\sqrt{t}}$ rate almost immediately by multiplicative weight update
- Lower bound per-step improvement based on smoothness of objective for fast rate (regret $\frac{1}{t}$)

3.2.7 Linear parameterization

Consider features $\phi(x,a)$, with linear softmax policies $\pi_\theta(a|x) \propto e^{\theta^T\phi(x,a)}$. Can we compete with the best linear policy, asssume $||\theta|| \leq 1, ||\phi(x,a) \leq 1||$?

$$\theta^* = \arg\max_{\theta} V^{\pi_\theta}(x_0)$$

NPG with linear parametrization, where the updates become

$$\theta^{t+1} = \theta^t + \eta w^t$$
$$\pi^{t+1}(a|x) \propto \pi^t(a|x)e^{\eta w^t\cdot\phi(x,a)}$$

The ideal w that NPG prefers, is solution to least squares problem defined as

$$w^t_* = \arg\min_{w} \underbrace{\mathbb{E}_{(x,a)\sim d^{(t)}}\Big[(Q^{(t)}(x,a) - w\cdot\phi(x,a))^2\Big]}_{L(w;\theta^t,d^{(t)})}$$

Using w^t_* is akin to running NPG. If we use some other w^t, we get some errror.
NPG with general parameterization

$$\theta^{t+1} = \theta^t + \eta w^t$$
$$\pi_\theta(a|x) \propto e^{f_\theta(x,a)}$$
$$w^t_* = \arg\min_{w} \mathbb{E}_{(x,a)\sim d^{(t)}}\Big[(Q^{(t)}(x,a) - w\cdot\nabla_\theta f_{\theta^t}(x,a))^2\Big]$$

where the gradients of $log\pi_\theta$ provides features for approximating values.
Convergence in linear case

- Estimation error: How well does w^t approximate w^t_*

$$\mathbb{E}\Big[L(w^t;\theta^t,d^{(t)}) - L(w^t_*;\theta^t,d^{(t)})\Big] \leq \epsilon_{stat}$$

- Transfer error under d*

$$\mathbb{E}\Big[L(w_*^t;\theta^t,d^*)\Big] \le \epsilon_{bias}$$

- Relative condition number : This condition holds if μ has full-rank covariance.

$$\frac{w^T\Sigma_{d^*}w}{w^T\Sigma_\mu w} \le \kappa \le \infty \qquad \text{where,} \quad \Sigma_d = \mathbb{E}_{x,a\sim d}\Big[\phi(x,a)\phi(x,a)^T\Big]$$

Theorem 8. *(AKLM19) with $\theta_0 = 0$ and $\eta = \sqrt{2log|\mathcal{A}|/T}$ we have*

$$V^*(x_0) - \max_t V^{(t)}(x_0) \le \underbrace{\sqrt{\frac{2log|\mathcal{A}|}{(1-\gamma)^2T}}}_{(i)} + \underbrace{\sqrt{\frac{4|\mathcal{A}|\kappa\epsilon_{stat}}{(1-\gamma)^3}}}_{(ii)} + \underbrace{\sqrt{\frac{4|\mathcal{A}|\epsilon_{bias}}{(1-\gamma)^2}}}_{(iii)}$$

- (i) we get a slower $\frac{1}{\sqrt{T}}$ rate as compared to tabular case (getting $\frac{1}{T}$ needs much stronger assumption)
- (ii) when $\epsilon_{stat} > 0$, depends on the initial exploration distribution through κ. show that irrespective of state, action size, as long as we embed in $R^d, \exists\mu$ s.t $\kappa \le d$. Finally, to fit w^t with N samples, we can achieve $\epsilon_{stat} = O(\frac{1}{\sqrt{N}})$
- (iii) note ϵ_{bias} is 0 if $Q^{(t)} = w_*^t \cdot \phi(x,a) \forall x,a$ (as in tabular, low-rank MDP).

Observations :

- with exact gradients, good representaitons, setting (ii) and (iii) to 0, we recover bounds $\sqrt{\frac{2log\mathcal{A}}{(1-\gamma)^2T}}$ which are similar to the tabular bounds.
- ϵ_{bias} gives a measure for quality of representations in capturing value of policy. this becomes increasingly involved in non-linear cases e.g NNs

3.2.8 Understanding the transfer errors

For relating error across two fitting distributions in regression like problems, one notion can be point-wise likelihood ratio reweighting. Especially in the batch setting with distribution $\mu(x,a)$

$$\begin{aligned} L(w_*^t;\theta_t,d^*) &\le \max_{x,a}\frac{d^*(x,a)}{d^{(t)}(x,a)}L(w_*^t;\theta_t,d^{(t)}) \\ &\le \max_{x,a}\frac{d^*(x,a)}{(1-\gamma)\mu(x,a)}L(w_*^t;\theta_t,d^{(t)}) \end{aligned}$$

Weakest conditions for RL with fixed μ (Sch14). The type dependence on density ratios in policy gradient methods is the nicest among other algorithms. First order optimization with small step sizies, updates the policy incrementally which seems to make it robust to modelling errors. Whenever the policy changes, we collect data which gives a good idea of the current state action distribution.

Question 2. *Conditions or assumptions to control the transfer error?*

3.3 Adding exploration

Under policies with linear parameterizations, we wan t to find exploratory distributions which have well conditioned feature covariance matrices with low relative condition number.
Key Idea : Find policies exploring different and use their mixture (say uniformly). How to identify such policies?

3.3.1 PC-PG

Algorithm 3 Policy-Cover Policy-Gradient

Policy cover : Set of exploratory policies discovered so far
for i **do** = 1, 2, ...
$\Pi_i \leftarrow$ current policy cover, $\rho_i \leftarrow Unif[\Pi_i]$
compute Σ_{ρ_i}, if $\phi(x,a) : \phi^T \Sigma^{-1} \phi \leq \kappa$ (it has good coverage)
mark x, a as known
Define bonus $b_i(x,a) = 1((x,a)$ is known)
$\pi^{i+1} = NPG(\rho_i, r + b_i)$
Update cover : $\Pi_{i+1} = \Pi_i \cup \pi^{i+1}$
end for

Theory for Low-Rank MDPs

- Let $\pi^* = \arg\max_{\pi \in \Pi_{linear}} V^\pi(s_0)$
- Let $\tilde{d}$ be the intrinsic dimension of the low-rank MDP (O(d) for finite dim)

  Theorem 9. *(AHKS20) For all low-rank MDPs in ϕ, PC-PG uses samples, computation poly($\tilde{d}, \frac{1}{\epsilon}$), $\frac{1}{1-\gamma}, W, ln\frac{1}{\delta}$) and w.p $\geq 1-\delta$ outputs policy π satisfying*

$$V^\pi(s_0) \geq V^{\pi^*}(s_0) - \epsilon$$

- Extends beyond the low-rank MDPs under transfer error conditions (CYJW19; ESRM20)
- Practically the algorithm is fairly modular. replace NPG with PPO, linear with neural policies.

Comparison with LSVI-UCB

- PC-PG handles infinite dimension.
- LSVI-UCB better data reuse, better sample complexity

$$\mathbb{E}_{x \sim d^*} \max_a \epsilon_{misspec}(x,a) \text{ versus } \max_{x,a} \epsilon_{misspec}(x,a)$$

3.4 Open questions

3.4.1 Better algorithms

- Can we quantify the benefits of optimization tricks? (includes variance reduction, acceleration, adaptive learnign rates, ...)
- Connections with online learing. Better potential functions?

3.4.2 Misspecified settings

- PG methods seem most stable to modeling assumption failures. (e.g transfer error)
- Isolated examples :
- Limits? Alternate notions of misspecification? (e.g. Q^* preserving aggregations)
- Is there fundamental tradeoff between robustness-efficiency? (e.g. with data reuse, stepsizes)

3.4.3 Exploration and Improvement

Always compete with a reference policy π_0, globally opt under assumptions. Suppose you need $f(\epsilon)$ samples to find globally ϵ-optimal policy when $\phi(x, a)$ induces a linear MDP. How to find a policy π using $f(\epsilon)$ samples, such that:

- either MDP is linear in ϕ and π is globally optimal, or
- $V(\pi) \geq V(\pi_0) - \epsilon^*$ (assuming $\pi_0 \in \Pi$, or we need other approximation terms)

4 Latent State Discovery

Consider MDP Learning + Generalization. Discover latent states to solve reinforcement learning. How do we use an oracle to efficiently solve an MDP?

4.1 The Block MDP Problem (DKJ+19)

Definition 4 (Block Markov Decision Process): *States $\mathcal{S}$, Actions $\mathcal{A}$, Initial States $P(s)$, Transition Matrix $T(s'|s, a)$, reward $R(r|s, a)$, Horizon H, Observations $q(x|s)$ with x disjoint over s (i.e the observation can decode the state through some unknown function f)*

- Assumption : Realizable supervised policy oracle

  For each cost sensitive classification learning problem $D(x, \overrightarrow{c})$ a policy class Π contains the optimal solutoin, and an oracle returns in unit time ($\overrightarrow{c}$ is cost for all actions (so we're in SL regime, not RL))

$$\arg\min_{\pi \in \Pi} \sum_{(x, \overrightarrow{c}) \in S} c_{\pi(x)}$$

4.1.1 Somethings that don't work in general

- Bottleneck autoencoder and declare the bottleneck a state (THF+16)

  Encoding "quality" is maximized by predicting pure noise bit.

- Inverse Kinematics : Predict previous action through bottleneck from previous and current observation (PAED17)

  Yet there exists a policy which can reach always

- Bisimulation (GDG03; LWL06): Alias "states" with same dynamics and rewards. Statistically intractable to learn (MJTS20)

4.2 HOMER (MHKL19)

Algorithm 4 Homer

for e **do**ach horizon h = 1 to H

Explore : Several times, fr wach latent state s reachable with $h-1$ steps, Use homing policy π_s to find observation x generated by latent state s. Use random action a, make next observation x'

Abstract : Lern to predict whether x' swapped

$$(p,\phi) = \arg\min_{p,\phi} \hat{\mathbb{E}}_{(x,a,x'),y}\Big(p(\phi(x),a,\phi(x')) - y\Big)^2$$

Home : For each value of bottleneck $s = \phi(x')$, learn homing policy π_s maximizing chance of s given homing policies

end for

return *Homing Policies*

- Homer results

  Theorem 10. $\forall$ *Block MDPs where every state can be visited with high probability, if learning oracles for* $\Pi, (p,\phi)$ *work with probability* $1-\delta$ *can learn* ε *optimal policy with* $poly(|\mathcal{A}|, |\mathcal{S}|, log|\mathcal{F}|, log|Pi|, log\frac{1}{\delta}, log\frac{1}{\varepsilon}, H)$ *samples.*

  Rich observations + deep learning $\implies$ covering set of policies

  covering set of policies $\implies$ efficiently learn to optimize any reward function

- Why does Homer work?

  Lemma 1. *If* x'_1, x'_2 *Backwards Kinematic Inseparable then* $\forall \pi_1, \pi_2 \frac{P_{\pi_1}(x'_1)}{P_{\pi_2}(x'_1)} = \frac{P_{\pi_1}(x'_2)}{P_{\pi_2}(x'_2)}$

Definition 5 (Backward Kinematic Inseparability): x'_1, x'_2 *are backwards KI if* $\forall u$ $\frac{T(x'_1|x,a)u(x,a)}{\sum_{\tilde{x},\tilde{a}} T(x'_1|\tilde{x},\tilde{a})u(\tilde{x},\tilde{a})} = \frac{T(x'_2|x,a)u(x,a)}{\sum_{\tilde{x},\tilde{a}} T(x'_2|\tilde{x},\tilde{a})u(\tilde{x},\tilde{a})}$

4.3 Latent State Decoding

Algorithm 5 general algorithm for latent state decoding

for episodes **do**

Explore : easily explored things

Abstract : latent state space

Cover : set of hard-to-reach things

end for

4.3.1 Learning latent low rank MDP? (AKKS20)

Algorithm 6 FLAMBE

for each horizon h=1 to H **do**
Explore : Several times, for each covering policy π execute for h-1 steps to observe x, then act randomly for a and observe x'
Abstract : maximize $\sum_{x,a,x'} log\phi(x,a)\mu(x')$
Cover : maximize range of $\phi(x,a)$ when finding new covering policies
end for

Theorem 11. $\forall$ *latent low rank MDPs, if learning oracles work, exploration succeeds after* $poly(|\mathcal{A}|, d, log|\phi|, log|M|, log\frac{1}{\delta}\frac{1}{\varepsilon}H)$ *samples.*

4.3.2 Learning linear dynamics? (MFS+20)

If it's possible to actually learn with liocal linear dynamics. Consider the cartpole problme wher the dynamics is $s' = As + Ba + \epsilon$. We want to solve this task based only the rich observations like images. So what should we do? In (MFS+20) the authors study recovering linear dynamics from non-linear observations.

Algorithm 7 RichID-CE

Explore Many times, act k times $a \sim N(0,1)$ and observe x
Abstract find $\hat{h} = \arg\min_h \sum_a (h(x) - a)^2, \hat{f}(x) = \hat{\mathbb{E}}\hat{h}(\overrightarrow{a}\hat{h}(\overrightarrow{a}^T\hat{h}(x))$
SystemID :$\hat{A}, \hat{B}, \ldots$
Improve optimal policy decoder

Theorem 12. $\forall$ *rich observation linear dynamics MDPs, if learning oracles work, exploration succeeds after* $poly(d_s, d_a, log|\mathcal{F}|, log\frac{1}{\delta}, \frac{1}{\varepsilon}, H)$ *samples*

5 Open Problems

5.1 Structural P1: Best definiton of latent state

Desirable properties

- Tractable to learn
- Parsimonius : The state abstraction should be as small as possible for tractability
- Sufficient for reward-optimizing policy

5.2 Structural P2: Other state-making prediction problems

- Low rank outcomes $>_{general}$ contrastive

- Inverse Kinematics?

  Works for linear dynamics, not MDPs.

- Is there a canonical way to make state from prediction problems?

5.3 Representational P1 : Tractably discover latent states

- Vast gap : tractable-in-theory and tractable-in-practice

5.4 Representational P2 : Other settings

- Linear dynamics + piecewise linear boundary conditions?
- Local linear dynamics?
- More general combinatorial state?

References

[ACBFS02] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The non-stochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[AHKS20] Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[AK08] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

[AKKS20] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in Neural Information Processing Systems*, 33, 2020.

[AKL20] Alekh Agarwal, Akshay Krishnamurthy, and John Langford. Tutorial on theoretical foundations of reinforcement learning. Foundations of Computer Science, 2020.

[AKLM19] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *arXiv preprint arXiv:1908.00261*, 2019.

[AYBB+19] Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702, 2019.

[BT02] Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

[CYJW19] Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830*, 2019.

[DKJ+19] Simon S Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudík, and John Langford. Provably efficient rl with rich observations via latent state decoding. *arXiv preprint arXiv:1901.09018*, 2019.

[DPWZ19] Kefan Dong, Jian Peng, Yining Wang, and Yuan Zhou. n-regret for learning in markov decision processes with function approximation and low bellman rank. 2019.

[EDKM09] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

[ESRM20] Yonathan Efroni, Lior Shani, Aviv Rosenberg, and Shie Mannor. Optimistic policy optimization with bandit feedback. *arXiv preprint arXiv:2002.08243*, 2020.

[GDG03] Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

[GSP19] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. *arXiv preprint arXiv:1901.11275*, 2019.

[HKM14] Elad Hazan, Zohar Karnin, and Raghu Meka. Volumetric spanners: an efficient exploration basis for learning. In *Conference on Learning Theory*, pages 408–422, 2014.

[HZAL18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[JAZBJ18] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

[JKA+17] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

[JOA10] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

[JYWJ20] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020.

[Kak01] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14:1531–1538, 2001.

[KL02] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

[KS02] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

[KT00] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[LCLS10] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[LWL06] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *ISAIM*, 2006.

[LZ07] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20:817–824, 2007.

[MFS+20] Zakaria Mhammedi, Dylan J Foster, Max Simchowitz, Dipendra Misra, Wen Sun, Akshay Krishnamurthy, Alexander Rakhlin, and John Langford. Learning the linear quadratic regulator from nonlinear observations. *Advances in Neural Information Processing Systems*, 33, 2020.

[MHKL19] Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. *arXiv preprint arXiv:1911.05815*, 2019.

[MJTS20] Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 2010–2020. PMLR, 2020.

[NJG17] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[PAED17] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[RVR13] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, pages 2256–2264, 2013.

[Sch14] Bruno Scherrer. Approximate policy iteration schemes: a comparison. In *International Conference on Machine Learning*, pages 1314–1322, 2014.

[SEM20] Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675, 2020.

[SJK+19] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933. PMLR, 2019.

[SLA+15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[SLW+06] Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888, 2006.

[SMSM99] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12:1057–1063, 1999.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[THF+16] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, FD Turck, and Pieter Abbeel. Exploration: A study of count-based exploration for deep reinforcement learning. corr abs/1611.04717 (2016). *arXiv preprint arXiv:1611.04717*, 2016.

[Tho33] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[WAS20] Gellert Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. *arXiv preprint arXiv:2010.01374*, 2020.

[WFK20] Ruosong Wang, Dean P Foster, and Sham M Kakade. What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*, 2020.

[Wil92] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[WSY20] Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33, 2020.

[ZLKB20] Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. *arXiv preprint arXiv:2003.00153*, 2020.