Towards a Theory of Generalization in RL

(Sham Kakade)

Challenges in RL

- Exploration (the environment is unknown)
- credit assignment problem → agent should be able to attribute what actions led to the reward / penalty.
- Large state / action spaces

Provable Generalization in RL?

- Can we find an ϵ-opt policy with no S-dependence, poly H and $\log(|\Pi|)$ dependence?
  ↳ set of policies

- Don't have to try all the policies independently in the world and we would like to reuse/re-utilize the data we have to do well. → possible to generalize without looking at all the configurations?

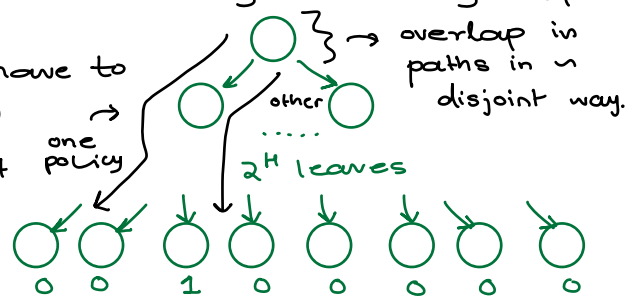
  ↳ No.

→ we need $\min(2^H, \log(|\Pi|)$ samples (for no S dependence)

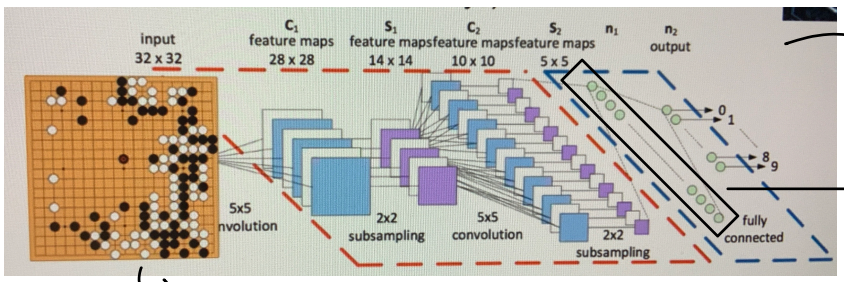
Proof: Consider a binary tree with a single rewarding leaf and 2^H policies.

- To find the best policy, we have to try 2^H policies (all of them) →
- → unlike SL, data reuse is not possible.

  ↳ intuition for challenge in generalization



- we can utilize some of the data but cannot reutilize to evaluate how well the other policy is doing?



↳ Some game configuration of Go

→ representations will help

→ transform the state into some feature mapping ϕ maybe that will help.

- Need strong assumptions to make the problem of RL retractable.

→ what are necessary conditions.

- What are the representational conditions under which generalization in RL is possible?

- What constitutes a good representation for RL?
  (i.e. permits sample efficient RL?) → good representations allow linear/log. representations to work (in ML).

widely used generalization: The linear bandit problem [Abe & Long+ '99]

Approx. DP with Linear Func. Approx.

Idea: Approx. the $Q(s,a)$ values with linear basis functions,

$$\underbrace{Q(s,a) = w \cdot \phi(s,a)}_{\text{linearly realizable } Q\text{-function.}}, \text{ where } \phi(s,a) \in \mathbb{R}^d \text{ and } d \ll |S||A|$$

→ having a representation and trying linear methods with this representation

An MDP is linearly realizable if there exists $w^* \in \mathbb{R}^d$ s.t for all (s,a)

$$Q_h^*(s,a) = w_h^* \cdot \phi(s,a)$$

(Q_h^*: optimal, linear in ϕ for every (s,a); ϕ: feature mapping)

Given the feature mapping, can you do sample efficient RL?

Theorem [Wang, Wang, K'21]: There exists a class of linearly realizable MDPs s.t any online RL algorithm requires $\min(\Omega(2^d), \Omega(2^H))$ samples to obtain a 0.1-near optimal policy (with prob. ≥ 0.9)

→ even if there is the feature mapping $\phi(s,a)$ perfectly representing (s,a) value, every online RL algorithm will require # of samples that's either exponential in dimension (d) or exponential in horizon (H) to find a near optimal policy.

- Results also hold even if the suboptimality gap is large:
  i.e. $\forall\ a \neq \pi^*(s)$, $\underbrace{V^*(s) - Q^*(s,a)}_{\geq \Delta_{min}}$

⇒ the theorem result is true even if the difference b/w best and second best action in every state is large.

The theorem also extends if the MDP has linearly realizable π^* if there exists w^* s.t. $\forall\ (s,a)$

$$\pi^*(s) = \arg\max_a w^* \cdot \phi(s,a)$$

→ results that every online RL algo. will require $\min(\Omega(2^d), \Omega(2^H))$ samples.

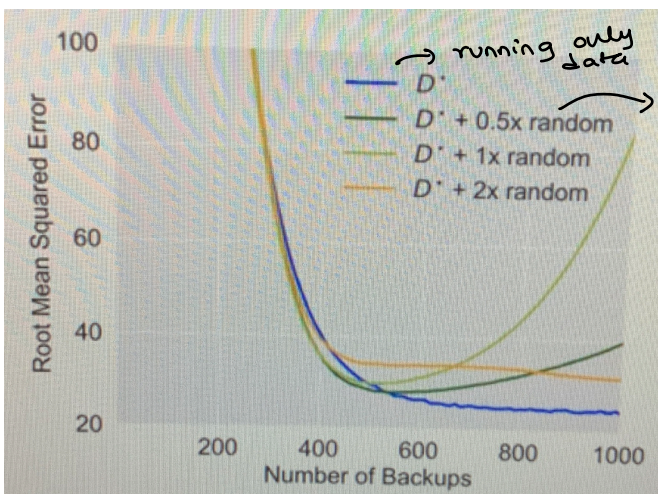
For π^*, the results hold even if there is a max. margin.

- The results show that linear realizability is not strong enough to make such assumptions.

Practical Off-line Setting

We have data coming from some policy (running) and some data from random step (policy) and the goal is to evaluate the running policy.
The features are from a pre-trained neural network to evaluate the running policy.
When the data is mixed from these two policies, we see error amplification on backup based methods.

↳ Even if the features are really good and they capture the policy under consideration, the error amplification will be there.



[Wang, Wu, Salakhutdinov, K. '21]
50% mixed offline data

- Offline dataset is mix of two sources running & random
- Use SL to evaluate the running policy with pre-trained features

→ Massive error amplification

What are the sufficient conditions

1) Linear Bellman Complete Classes

Linear hypothesis class: $\mathcal{F} = \{Q_f : Q_f(s,a) = w(f) \cdot \phi(s,a)\}$ (features; linear mapping)

Completeness: $\mathcal{F}$ is closed under Bellman backups i.e.

$\mathcal{T}(Q_f) \in \mathcal{F}$ where $\mathcal{T}(Q_f)(s,a) = r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)}[\max_{a'} Q(s',a')]$ (take a; greedy action)

Bellman optimality: $Q^* - \mathcal{T}(Q^*) = 0$

↳ Start with Q, apply $\mathcal{T}(\cdot)$ to it and it stays in hypothesis class

Completeness is strong assumption:
- Adding features to ϕ can break the completeness property

⇒ Sample efficient RL, poly $(d, H, 1/\varepsilon)$ is possible with Bellman complete, linear $\mathcal{F}$.

- Key property allowing data re-use: Bellman error of f is estimable using an π.

$$E_\pi\left[Q_f(s_h,a_h) - r(s_h,a_h) - V_f(s_{h+1})\right] \leq \left\langle w(f) - \mathcal{J}(w(f)),\ E_\pi\left[\phi(s_h,a_h)\right]\right\rangle$$

linear func.

greedy value of Q_f

upper bounded the linear function

- For any policy π, we can evaluate the Bellman error of the function f.

→ Bellman error of some hypothesis f when using another policy π, its linear.

Bilinear Regret Classes

Hypothesis class $\{f \in \mathcal{F}\}$

Def: A $(\mathcal{F}, \mathcal{L})$ forms an implicit Bilinear class if

- Bilinear regret: on-policy difference b/w claimed & true reward

$$\left|E_\pi\left[Q_f(s,a) - r(s_h,a_h) - V_f(s_{h+1})\right]\right| \leq \left\langle w_h(f) - w_h^*,\ \phi_h(f)\right\rangle$$

- Data re-use: there is a function $\ell_f(s,a,s',g)$ s.t. (discrepancy function)

$$E_{\pi_f}\left[\ell_f(s_h,a_h,s_{h+1},g)\right] = \left\langle w_h(g) - w_h^*,\ \phi_h(f)\right\rangle$$