Statistical Limits of Generalization

Sham M. Kakade and Wen Sun

Outline

1. Recap: Sample Complexity
2. Today: SL vs. RL
3. Supervised Learning (SL) : Let's review
4. RL and generalization
5. Linear Realizability

Minimax Optimal Sample Complexity (on the policy)

using the model based approach

Theorem: (Agarwal et al. '20) For $\epsilon < \sqrt{1/(1-\gamma)}$, provided

$$N \geq \frac{c}{(1-\gamma)^3} \frac{\log(cSA/\delta)}{\epsilon^2}$$

then with prob. greater than $1-\delta$),

$$\|Q^\star - Q^{\hat{\pi}\star}\|_\infty \leq \epsilon$$

total samples is $O\left(\frac{SA}{(1-\gamma)^3 \epsilon^2}\right)$

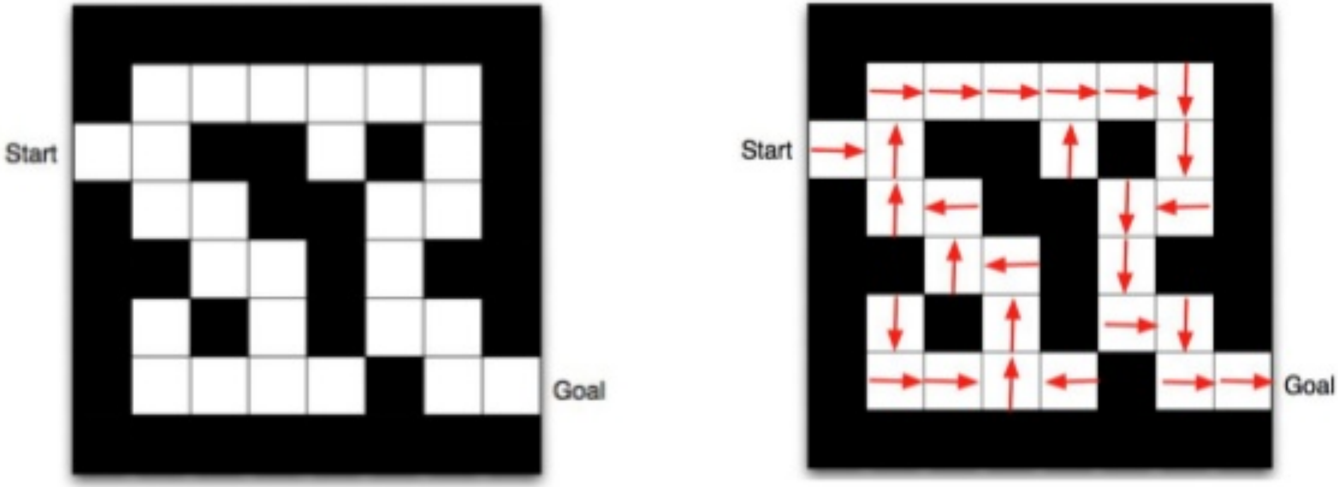
Lower Bound: We can't do better.

Outline

1. Recap: Sample Complexity
2. Today: SL vs. RL
3. Supervised Learning (SL) : Let's review
4. RL and generalization
5. Linear Realizability

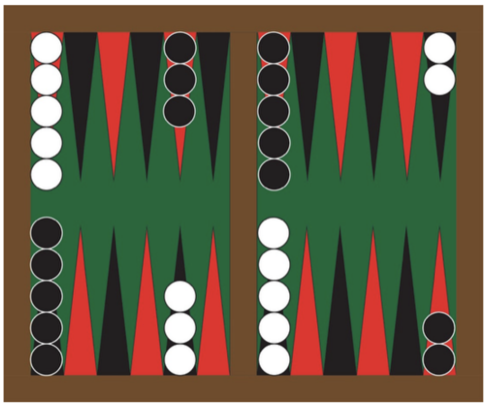
What we have studied so far: (the small state space case)

Maze example: r = -1 per time-step and policy



[David Silver. Advanced Topics: RL]

What we want to solve:
(the large state space case)



TD GAMMON [Tesauro 95]

[AlphaZero, Silver et.al, 17]

[OpenAI Five, 18]

Generalization: RL vs Supervised Learning (SL)

To what extent is generalization in RL similar to (or different from) that in supervised learning?

Generalization: RL vs Supervised Learning (SL)

To what extent is generalization in RL similar to (or different from) that in supervised learning?

- We seek to learn when the number of states is large (or infinite). This is a question of generalization.

Generalization: RL vs Supervised Learning (SL)

To what extent is generalization in RL similar to (or different from) that in supervised learning?

- We seek to learn when the number of states is large (or infinite). This is a question of generalization.
- Supervised Learning: two important frameworks
  - Agnostic learning: try to do as well best classifier in some (restricted) class $\mathcal{H}$.
  - Linear models: learn the best linear regressor or binary classifier (among halspaces)

Generalization: RL vs Supervised Learning (SL)

To what extent is generalization in RL similar to (or different from) that in supervised learning?

- We seek to learn when the number of states is large (or infinite). This is a question of generalization.
- Supervised Learning: two important frameworks
  - Agnostic learning: try to do as well best classifier in some (restricted) class $\mathcal{H}$.
  - Linear models: learn the best linear regressor or binary classifier (among halspaces)
- Reinforcement Learning: analogous questions
  - Agnostic learning: can we find the best policy in some (restricted) class Π (rather than trying to be optimal)?
  - Linear realizability: if the optimal value or policy is parameterized with a linear model, can we learn with fewer samples?

Outline

1. Recap: Sample Complexity
2. Today: SL vs. RL
3. Supervised Learning (SL) : Let's review
4. RL and generalization
5. Linear Realizability

Binary Classification

- n labeled examples: $(x_i, y_i)_{i=1}^{n}$, with $x_i \in \mathcal{X}$ and $y_i \in \{0, 1\}$.
  A set $\mathcal{H}$ of binary classifiers, where for $h \in \mathcal{H}$, $h : \mathcal{X} \to \{0, 1\}$.
  Define the empirical error and the true error as:

$$\widehat{\text{err}}(h) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(h(x_i) \neq y_i), \quad \text{err}(h) = \mathbb{E}_{(X,Y)\sim D} \mathbf{1}(h(X) \neq Y).$$

where $\mathbf{1}(h(x) \neq y)$ is 0 if $h(x) = y$ and 1 otherwise.

Binary Classification

- n labeled examples: $(x_i, y_i)_{i=1}^n$, with $x_i \in \mathcal{X}$ and $y_i \in \{0, 1\}$.
  A set $\mathcal{H}$ of binary classifiers, where for $h \in \mathcal{H}$, $h : \mathcal{X} \to \{0, 1\}$.
  Define the empirical error and the true error as:

$$\widehat{\mathrm{err}}(h) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(h(x_i) \neq y_i), \quad \mathrm{err}(h) = \mathbb{E}_{(X,Y)\sim D} \mathbf{1}(h(X) \neq Y).$$

  where $\mathbf{1}(h(x) \neq y)$ is 0 if $h(x) = y$ and 1 otherwise.
- If the samples are drawn i.i.d. according to a joint distribution D over (x, y), then, by Hoeffding's inequality, for a fixed $h \in \mathcal{H}$, with probability at least $1 - \delta$:

$$|\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \leq \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}.$$

SL is an RL problem with $\gamma = 0$

$H = 1$

- Binary classification is special case of RL.
  Consider learning in an MDP, with two actions where the effective horizon is 1.
- $|\mathcal{A}| = 2$, $\gamma = 0$, and the reward function is $r(s, a) = \mathbf{1}(\mathrm{label}(s) = a)$.
- Note in SL, we rarely make restrictions that $\mathcal{X}$ (i.e. $\mathcal{S}$) is finite.
- Note that $\mu(s_0) \leftrightarrow D(x)$ (D is the distribution of our data)

Occams Razor and Generalization

Your HW0: This and the union bound give rise to what is often referred to as the “Occam’s razor” bound:

Proposition

(The “Occam’s razor” bound) Suppose $\mathcal{H}$ is finite. Let $\widehat{h} = \arg\min_{h\in\mathcal{H}} \widehat{err}(h)$ and $h^\star = \arg\min_{h\in\mathcal{H}} err(h)$. With probability at least $1-\delta$:

$$err(\widehat{h}) - err(h^\star) \le \sqrt{\frac{2}{N}\log\frac{2|\mathcal{H}|}{\delta}}.$$

Occams Razor and Generalization

Your HW0: This and the union bound give rise to what is often referred to as the “Occam’s razor” bound:

Proposition

(The “Occam’s razor” bound) Suppose $\mathcal{H}$ is finite. Let $\widehat{h} = \arg\min_{h\in\mathcal{H}} \widehat{err}(h)$ and $h^\star = \arg\min_{h\in\mathcal{H}} err(h)$. With probability at least $1-\delta$:

$$err(\widehat{h}) - err(h^\star) \le \sqrt{\frac{2}{N}\log\frac{2|\mathcal{H}|}{\delta}}.$$

(The logarithmic dependence is the most naive complexity measure of $\mathcal{H}$, yet the bound is strong.)

Outline

1. Recap: Sample Complexity
2. Today: SL vs. RL
3. Supervised Learning (SL) : Let's review
4. RL and generalization
5. Linear Realizability

Finite Horizon MDPs

- finite horizon, time-dependent Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \{P\}_h, \{r\}_h, H, \mu)$ is specified as follows:

Finite Horizon MDPs

- finite horizon, time-dependent Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \{P\}_h, \{r\}_h, H, \mu)$ is specified as follows:
  - A integer H which defines the horizon of the problem.

Finite Horizon MDPs

- finite horizon, time-dependent Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \{P\}_h, \{r\}_h, H, \mu)$ is specified as follows:
  - A integer H which defines the horizon of the problem.
  - A time-dependent transition function: for $h \in [H]$, $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$

Finite Horizon MDPs

- finite horizon, time-dependent Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \{P\}_h, \{r\}_h, H, \mu)$ is specified as follows:
  - A integer H which defines the horizon of the problem.
  - A time-dependent transition function: for $h \in [H]$, $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$
  - A time-dependent reward function: for $h \in [H]$, $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$.

Finite Horizon MDPs

undiscounted

- finite horizon, time-dependent Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \{P\}_h, \{r\}_h, H, \mu)$ is specified as follows:
  - A integer H which defines the horizon of the problem.
  - A time-dependent transition function: for $h \in [H]$, $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$
  - A time-dependent reward function: for $h \in [H]$, $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$.
- Goal:

$$\text{argmax}_\pi \, E_{s_0 \sim \mu} V^\pi(s_0), \quad \text{where } V^\pi(s_0) = \mathbb{E}\Big[\sum_{t=0}^{H-1} r_h(s_t, a_t) \;\Big|\; \pi, s_0\Big]$$

$0 \leq \quad \leq H$

Bellman equations: finite horizon case

- Define the value functions $V_h^\pi : \mathcal{S} \to \mathbb{R}$ as

$$V_h^\pi(s) = \mathbb{E}\Big[\sum_{t=h}^{H-1} r_h(s_t, a_t) \;\Big|\; \pi, s_h = s \Big],$$

Bellman equations: finite horizon case

- Define the value functions $V_h^\pi : \mathcal{S} \to \mathbb{R}$ as

$$V_h^\pi(s) = \mathbb{E}\Big[\sum_{t=h}^{H-1} r_h(s_t, a_t) \;\Big|\; \pi, s_h = s\Big],$$

- Define the state-action value function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$Q_h^\pi(s, a) = \mathbb{E}\Big[\sum_{t=h}^{H-1} r_h(s_t, a_t) \;\Big|\; \pi, s_h = s, a_h = a\Big].$$

Bellman equations: finite horizon case

- Define the value functions $V_h^\pi : \mathcal{S} \to \mathbb{R}$ as

$$V_h^\pi(s) = \mathbb{E}\Big[\sum_{t=h}^{H-1} r_h(s_t, a_t) \,\Big|\, \pi, s_h = s\Big],$$

- Define the state-action value function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$Q_h^\pi(s, a) = \mathbb{E}\Big[\sum_{t=h}^{H-1} r_h(s_t, a_t) \,\Big|\, \pi, s_h = s, a_h = a\Big].$$

- Bellman optimality equations: Define $Q_h^\star(s, a) = \sup_{\pi\in\Pi} Q_h^\pi(s, a)$. Suppose that $Q_H = 0$. We have that $Q_h = Q_h^\star$ for all $h \in [H]$ if and only if for all $h \in [H]$,

$$Q_h(s, a) = r_h(s, a) + \gamma\mathbb{E}_{s'\sim P_h(\cdot|s,a)}\left[\max_{a'\in\mathcal{A}} Q_{h+1}(s', a')\right].$$

Furthermore, $\pi(s, h) = \text{argmax}_{a\in\mathcal{A}}\, Q_h^\star(s, a)$ is an optimal policy.

RL and Agnostic Learning

- We have a set of policies Π (either finite or infinite).
  - Π could be a parametric set.
  - Π could be greedy policys on a a set of parametric value functions $\mathcal{V} = \{f_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} | \theta \in \mathbb{R}^d\}$. set of value functions
  - Π may not contain $\pi^\star$.

Π = set of greedy policies for $f \in \mathcal{V}$

RL and Agnostic Learning

- We have a set of policies Π (either finite or infinite).
  - Π could be a parametric set.
  - Π could be greedy policys on a a set of parametric value functions $\mathcal{V} = \{f_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} | \theta \in \mathbb{R}^d\}$.
  - Π may not contain $\pi^\star$.
- in agnostic learning, we have the optimization problem:

$$\max_{\pi \in \Pi} V^\pi(s_0)$$

We want to (approx) solve this with a small number of sample trajectories.

RL and Agnostic Learning

- We have a set of policies Π (either finite or infinite).
  - Π could be a parametric set.
  - Π could be greedy policys on a a set of parametric value functions $\mathcal{V} = \{f_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} | \theta \in \mathbb{R}^d\}$.
  - Π may not contain $\pi^\star$.
- in agnostic learning, we have the optimization problem:

$$\max_{\pi \in \Pi} V^\pi(s_0)$$

  We want to (approx) solve this with a small number of sample trajectories.
- analogous to agnostic learning in SL
  - binary classification: $|\mathcal{A}| = 2$, $\cancel{\gamma = 0}$, $r(\cdot)$ being the labeling reward.
  - relevant dependencies for RL: $|H|=1$??

$$\text{Complexity}(\Pi), |\mathcal{S}|, |\mathcal{A}|, N, H$$

Episodic Model

$\mu = S_0$

- Assume sampling access to the MDP in a μ-reset model:
  - start at a state $s_0 \sim \mu$
  - we can rollout a policy π of our choosing
  - we can terminate the trajectory at will.

  (weaker model than generative model)

How can we "reuse" data to do agnostic learning?

Importance Sampling

- Let $\mathrm{Unif}_{\mathcal{A}}$ be the uniformly random policy.
- Using $\mathrm{Unif}_{\mathcal{A}}$ in the episodic model can provide an unbiased estimate of any other policy π.

why? importance sampling

Lemma

(Unbiased estimation of $V_0^\pi(s_0)$) For any deterministic policy π,

$$V_0^\pi(s_0) =$$

$$|\mathcal{A}|^H \mathbb{E}_{a_{0:H-1}\sim Unif_{\mathcal{A}}}\left[\mathbf{1}\Big(\pi(s_0)=a_0,\ \ldots,\pi(s_H)=a_H\Big)\sum_{t=0}^{H-1} r(s_t,a_t)\Big|s_0\right]$$

(note that the expectation is with respect to trajectory generated by following the actions under $Unif_{\mathcal{A}}$).

An Occams Razor Bound for RL

- Collect N trajectories with $\mathrm{Unif}_{\mathcal{A}}$.

An Occams Razor Bound for RL

- Collect N trajectories with $\mathrm{Unif}_{\mathcal{A}}$.
- Denote the n-th sample by $(s_0^n, a_0^n, r_1^n, s_1^n, \ldots, s_{H-1}^n)$.

An Occams Razor Bound for RL

- Collect N trajectories with $\mathrm{Unif}_{\mathcal{A}}$.
- Denote the n-th sample by $(s_0^n, a_0^n, r_1^n, s_1^n, \ldots, s_{H-1}^n)$.
- Consider the following empirical estimator of any π:

$$\widehat{V}^{\pi}(s_0) = \frac{|\mathcal{A}|^H}{N} \sum_{n=1}^{N} \mathbf{1}\Big(\pi(s_0^n) = a_0^n, \ldots \pi(s_{H-1}^n) = a_{H-1}^n\Big) \sum_{t=0}^{H-1} r(s_t^n, a_t^n).$$

An Occams Razor Bound for RL

No dependence on S.

- Collect N trajectories with $\mathrm{Unif}_{\mathcal{A}}$.
- Denote the n-th sample by $(s_0^n, a_0^n, r_1^n, s_1^n, \ldots, s_{H-1}^n)$.
- Consider the following empirical estimator of any π:

$$\widehat{V}^\pi(s_0) = \frac{|\mathcal{A}|^H}{N} \sum_{n=1}^{N} \mathbf{1}\Big(\pi(s_0^n) = a_0^n, \ldots \pi(s_{H-1}^n) = a_{H-1}^n\Big) \sum_{t=0}^{H-1} r(s_t^n, a_t^n).$$

Proposition

(Generalization in RL) Suppose Π *is finite. Let* $\widehat{\pi} = \arg\max_{\pi \in \Pi} \widehat{V}^\pi(s_0)$*. With probability at least* $1 - \delta$*:*

$$V^{\widehat{\pi}}(s_0) \geq \arg\max_{\pi \in \Pi} V^\pi(s_0) - H|\mathcal{A}|^H \sqrt{\frac{2}{N} \log \frac{2|\Pi|}{\delta}}.$$

Can we do better?

- What we want, for an agnostic sample complexity:
  - no dependence on $|\mathcal{S}|$ (or logarithmic)
  - poly H dependence
  - to depend reasonably on a complexity measure of $\mathcal{H}$
    e.g. poly $\log|\mathcal{H}|$ dependence

Can we do better?

- What we want, for an agnostic sample complexity:
  - no dependence on $|\mathcal{S}|$ (or logarithmic)
  - poly H dependence
  - to depend reasonably on a complexity measure of $\mathcal{H}$
    e.g. poly $\log|\mathcal{H}|$ dependence
- Is this possible?

Can we do better?

- What we want, for an agnostic sample complexity:
  - no dependence on $|\mathcal{S}|$ (or logarithmic)
  - poly H dependence
  - to depend reasonably on a complexity measure of $\mathcal{H}$
    e.g. poly $\log|\mathcal{H}|$ dependence
- Is this possible?
  No :(

Can we do better?

- What we want, for an agnostic sample complexity:
  - no dependence on $|\mathcal{S}|$ (or logarithmic)
  - poly H dependence
  - to depend reasonably on a complexity measure of $\mathcal{H}$
    e.g. poly $\log|\mathcal{H}|$ dependence
- Is this possible?
  No :(
- This is (one reason) why RL is challenging.
  (both in theory and in practice)

An "Easy" Lower Bound

Proposition

(Lower Bound) Suppose $\mathcal{A}$ has access to a generative model. There exists a policy class Π, with $|\Pi| = |\mathcal{A}|^H$ such that if $\mathcal{A}$ returns a policy π where

$$V_0^\pi(\mu) \geq \arg\max_{\pi\in\Pi} V_0^\pi(\mu) - 0.5.$$

with probability greater than $1/2$, then $\mathcal{A}$ use a number of samples:

$$N \geq c|\mathcal{A}|^H$$

$\log |\pi|$

$= H \cdot \log |A|$

(where c is an absolute constant).

An “Easy” Lower Bound

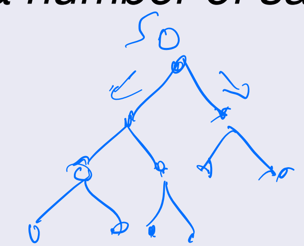
Proposition

(Lower Bound) Suppose $\mathcal{A}$ has access to a generative model. There exists a policy class Π, with $|\Pi| = |\mathcal{A}|^H$ such that if $\mathcal{A}$ returns a policy π where

$$V_0^\pi(\mu) \geq \arg\max_{\pi \in \Pi} V_0^\pi(\mu) - 0.5.$$

with probability greater than $1/2$, then $\mathcal{A}$ use a number of samples:

$$N \geq c|\mathcal{A}|^H$$



$|\mathcal{A}| = 2$

$\{$left, right$\}$

(where c is an absolute constant).

one leaf with $r = 1$.

Proof: Consider a full $|\mathcal{A}|$-ary tree of depth H, which defines the MDP. Suppose there is only one rewarding leaf node. There are $|\mathcal{A}|^H$ deterministic policies. And we require $\Omega(|\mathcal{A}|^H)$ queries to discover the rewarding leaf node.

$|\Pi| = 2^H$

Outline

1. Recap: Sample Complexity
2. Today: SL vs. RL
3. Supervised Learning (SL) : Let's review
4. RL and generalization
5. Linear Realizability