A Brief Introduction to Theoretical Foundations of Machine Learning and Machine Teaching

Jerry Zhu

University of Wisconsin-Madison

Simons Workshop on Synthesis of Models and Systems
3/15/2021

Outline

Hypothesis Space

- X: input space, e.g. natural numbers $\mathbb{N}$ (in general $\mathbb{R}^d$)
- Y: output space, e.g. $\{0, 1\}$
- $h : X \mapsto Y$: a hypothesis, e.g. $h_i(x) = \mathbb{1}[x \geq i]$ or $h_i = 0 \ldots 0111111 \ldots$
- $\mathcal{H} \subseteq Y^X$: hypothesis space, e.g. $\mathcal{H} = \{h_i : i \in \mathbb{N}\}$
- target $h^* \in Y^X$
  - $h^* \in \mathcal{H}$: realizable, e.g. h_{2021}
  - $h^* \notin \mathcal{H}$: agnostic, e.g. $h^* = 10111111 \ldots$

Passive Learning Protocol

- Environment has $P(x, y)$, e.g.
  - $P(x) = \lambda(1-\lambda)^{x-1}$
  - $P(y \mid x) = \mathbb{1}[y = h^*(x)]$
- Environment draws training set $S = (x_1, y_1) \dots (x_n, y_n) \overset{iid}{\sim} P(x, y)$
  - Example 1: $h^* = h_{2021}$, modest n
  - S may not contain large x values.
  - Say $\max_{i=1}^n x_i = 100$, then $y_1 = \dots = y_n = 0$
- Learner receives S and selects $\hat{h} \in \mathcal{H}$
  - In Example 1 $\hat{h}$ can be h_{101}, very different from h^*
  - But this is OK since machine learning only cares about the risk

True Risk and Empirical Risk

- Loss $\ell(y, y') \geq 0$, e.g. 0-1 loss $\mathbb{1}[y \neq y']$
- True risk $R(h) = \mathbb{E}_P(\ell(h(x), y))$
  - How $P(x, y)$ relates to h^*: $h^* = \operatorname{argmin}_{h \in \mathcal{Y}^X} R(h)$
  - Learner's goal is small $R(\hat{h})$, not $\hat{h} = h^*$
  - Test set error is a Monte Carlo estimate of R
- Empirical risk (training set error) on S: $\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$

Empirical Risk Minimization (ERM)

- Learner wants to minimize R, but only observes $\hat{R}$
- ERM is a learning algorithm:

$$\hat{h} \in \operatorname*{argmin}_{h \in \mathcal{H}} \hat{R}(h) = \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$$

- In Example 1 the argmin set is $\{h_{101}, h_{102}, \ldots\}$
- The learned ERM $\hat{h}$ can be any one of them

Overfitting

Overfitting is a non-technical term, could mean

- $R(\hat{h}) \gg \hat{R}(\hat{h})$, "my test error is much higher than training set error"
- $R(\hat{h}) \gg R(h^*)$, "I didn't get the best risk"
- $R(\hat{h}) \gg \inf_{h' \in \mathcal{H}} R(h')$, "I didn't get the best risk even among the models available to me"

Risk Decomposition

$$
\begin{aligned}
R(\hat{h}) \quad = \quad & \left[R(\hat{h}) - \inf_{h' \in \mathcal{H}} R(h') \right] \ \text{estimation error} \\
& + \left[\inf_{h' \in \mathcal{H}} R(h') - R(h^*) \right] \ \text{approximation error} \\
& + [R(h^*)] \ \text{Bayes error}
\end{aligned}
$$

Example 2: $\mathcal{H} = \{h_i = 0 \ldots 0111111 \ldots : i \in \mathbb{N}\}$, $h^* = 10111111 \ldots$

- Bayes error: $P(y \mid x)$ not concentrated on $y = h^*(x)$
- approximation error: $h^* \notin \mathcal{H}$, closest to $\arg\inf_{h' \in \mathcal{H}} R(h') = h_1 = 111111 \ldots$ under geometric $P(x)$
- estimation error: $S \sim P^n(x, y)$ is finite and random. If S contains $x = 2$ but not $x = 1$, ERM will pick $\hat{h} = h_3$

Probably-Approximately-Correct (PAC) Guarantee

Assume finite $\mathcal{H}$.

Theorem

For any $\delta > 0$

$$P_S\left(R(\hat{h}) - \inf_{h' \in \mathcal{H}} R(h') \leq \sqrt{\frac{2}{n}\log\frac{2|\mathcal{H}|}{\delta}}\right) \geq 1 - \delta$$

- You probably will not receive a strange S
- Under typical S estimation error <u>bound</u> decreases as $O(\frac{1}{\sqrt{n}})$
- Can sharpen to $O(\frac{1}{n})$ for realizable case
- No control over approximation and Bayes errors

Probably-Approximately-Correct (PAC) Guarantee

How we get there:

1. Fixing h, $|R(h) - \hat{R}(h)| \lesssim \frac{1}{\sqrt{n}}$ by Hoeffding's inequality (just Monte Carlo)
2. Uniform convergence $\forall h \in \mathcal{H} : |R(h) - \hat{R}(h)| \lesssim \sqrt{\frac{\log |\mathcal{H}|}{n}}$ by a union bound
3. $\hat{h}$ chosen by ERM: $\hat{R}(\hat{h}) \leq \hat{R}(\text{best } h' \in \mathcal{H})$
4. $\Rightarrow R(\hat{h})$ cannot be much larger than $R(\text{best } h' \in \mathcal{H})$

Vapnik-Chervonenkis (VC) Dimension

- Recall our $\mathcal{H} = \{h_i = 0 \ldots 0111111 \ldots : i \in \mathbb{N}\}$: $|\mathcal{H}| = \infty$
- Should be learnable: union bound too weak!
- $VC(\mathcal{H})$: size t of the largest set $\{x_{i_1}, \ldots, x_{i_t}\}$ that can be assigned all 2^t labels by $\mathcal{H}$ (shattering)
  - $t = 1$: $\{x = 1\}$ assigned label 0 by h_2, label 1 by h_1
  - $t = 2$: $\{x = 1, x = 2\}$ assigned labels 00 by h_3, labels 01 by h_2, labels 11 by h_1, but not 10
  - No $x_1 < x_2$ can be assigned 10 by $\mathcal{H}$
  - Our $VC(\mathcal{H}) = 1$

PAC Guarantee, Revisited

(Previously) finite $\mathcal{H}$: with probability at least $1-\delta$,

$$R(\hat{h}) - \inf_{h' \in \mathcal{H}} R(h') \leq O\left(\sqrt{\frac{\log|\mathcal{H}| + \log 1/\delta}{n}}\right)$$

Theorem

Finite $VC(\mathcal{H})$: with probability at least $1-\delta$,

$$R(\hat{h}) - \inf_{h' \in \mathcal{H}} R(h') \leq O\left(\sqrt{\frac{VC(\mathcal{H}) + \log 1/\delta}{n}}\right)$$

Passive Learning Summary

- Environment draws training set $S = (x_1, y_1) \ldots (x_n, y_n) \overset{iid}{\sim} P(x, y)$
  - Learner has no say in data
  - Environment is not particularly helpful
- When $VC(\mathcal{H}) < \infty$, estimation error bound $O(\frac{1}{\sqrt{n}})$
  - approximation and Bayes errors uncontrolled
  - deep learning requires additional theory, active research area

Outline

Passive Learning (PAC Learning, Statistical Learning, Learning from iid Data)

Active Learning

Machine Teaching: Helpful Teachers

Online Learning

Multi-Armed Bandits

Reinforcement Learning

For Simplicity...

We will assume

- no Bayes error: $P(y = h^*(x) \mid x) = 1$
- no approximation error: $h^* \in \mathcal{H}$

Both can be relaxed.

Active Learning Protocol

$\mathcal{H}$ is common knowledge. Environment has $h^* \in \mathcal{H}$.

1. For $t = 1, 2, \ldots$
2. learner asks query $x_t \in X$ based on history
3. oracle answers label $y_t = h^*(x_t)$
4. learner estimates $\hat{h}_t \in \mathcal{H}$

Two flavors of query x_t:

- learner synthesizes any $x \in X$ (the Membership Query of [Angluin'88] is a special case for binary Y)
- learner repeatedly draws $x \sim P(x)$ until it likes the x (assuming unlabeled data costs nothing)

Example: Binary Search

Example 3:

- $X = [0, 1], P(x) = \text{uniform}(X), Y = \{0, 1\}$
- $h_a(x) = \mathbb{1}[x \geq a], \mathcal{H} = \{h_a : a \in X\}$
- h^* has threshold $a^* \in X$
- Query x_t by binary search over X

Binary Search Analysis

- After n queries, the interval containing a^* has length

$$1/2^n$$

- Pick any $\hat{h}_t$ in that interval
- $R(\hat{h}_t) \leq 1/2^n$ (recall $P(x) = \mathrm{uniform}[0,1]$)
- Exponential speed up compared to passive learning's $R(\hat{h}_t) = O(1/n)$

Beyond Binary Search

- Nice, but only works for threshold functions.
- New concepts
  - version space

$$V = \{h \in \mathcal{H} : h \text{ agrees with all data seen so far}\}$$

  - disagreement region

$$DIS(V) = \{x \in X : \exists h, h' \in V, h(x) \neq h'(x)\}$$

CAL: A General Active Learning Algorithm

Assume $|\mathcal{H}| < \infty$, realizable

1. Version space $V = \mathcal{H}$
2. While $P(DIS(V)) \geq \epsilon$
3. repeat $x \sim P(X)$ until we have k points in $DIS(V)$
4. query these k points
5. $V \leftarrow \{h \in V : h$ agrees with these k points$\}$
6. Output any $\hat{h} \in V$

Intuition: In iteration i, k random points in $DIS(V_i)$ reduce V_i's radius $r(V_i) = \max_{h \in V_i} R(h)$ by at least half.

CAL Guarantee

Let $k = 2\theta\left(\log\frac{|\mathcal{H}|}{\delta} + \log\log\frac{1}{\epsilon}\right)$ in step 3.

Theorem

With probability at least $1-\delta$, CAL terminates after $\log\frac{1}{\epsilon}$ iterations, and $R(\hat{h}) \le \epsilon$. The number of queries is

$$O\left(\left(\log\frac{1}{\epsilon}\right)\theta\left(\log\frac{|\mathcal{H}|}{\delta} + \log\log\frac{1}{\epsilon}\right)\right).$$

- Number of queries $n = O\left(\log\frac{1}{\epsilon}\right)$ implies $R(\hat{h}) = O\left(1/e^n\right)$
- Depends on θ being small

Disagreement Coefficient θ

$$\theta = \sup_{r \in (0,1)} \frac{P(DIS(\mathbb{B}(h^*, r)))}{r}$$

- $\mathcal{H}$ = 1D thresholds
  - $h^* = h_{a^*}$
  - $\mathbb{B}(h^*, r) = \{h_a : a \in [a^* - r, a^* + r]\}$
  - $DIS(\mathbb{B}(h^*, r)) = \{x : a^* - r \le x \le a^* + r\}$
  - $P(DIS(\mathbb{B}(h^*, r))) = 2r$
  - $\theta = \sup_{r \in (0,1)} \frac{P(DIS(\mathbb{B}(h^*, r)))}{r} = 2$
- $\mathcal{H}$ = 1D intervals $[a^*, b^*]$
  - $\theta = \max\left(\frac{1}{\max(b^* - a^*, \epsilon)}, 4\right)$
  - trouble when $b^* - a^*$ small
  - "warm start" problem (hit the interval) of active learning
- $\mathcal{H}$ = d-dim hyperplane $\mathbb{1}[\mathbf{w}^\top \mathbf{x} + b \ge 0]$: $\theta = O(1)$ under mild conditions on $P(x, y)$

Active Learning Summary

- Learner queries x_t
- Environment answers $h^*(y_t)$
- CAL error bound $O(e^{-\frac{n}{\theta}})$
- Potential exponential speed-up due to freedom in choosing x

Active Learning with Equivalence Queries?

1. For $t = 1, 2, \ldots$
2. learner asks equivalence query $\hat{h}_{t-1} \in \mathcal{H}$
3. oracle answers "yes" or counterexample $\left(x_t \in DIS(h^*, \hat{h}_{t-1}), y_t = h^*(x_t)\right)$
4. learner estimates $\hat{h}_t \in \mathcal{H}$

- Not well-studied in machine learning
- In classic work x_t is adversarial (least helpful oracle)
- But we can imagine a helpful oracle...

Helpful Oracle on Equivalence Queries

Recall Example 1: $\mathcal{H} = \{h_i = 0 \ldots 0111111 \ldots : i \in \mathbb{N}\}$, $h^* = h_{2021}$

- Least-helpful oracle
  - query: $\hat{h} = 111111\ldots$?
  - answer: no. $(x = 1, y = 0)$
  - query: $\hat{h} = 011111\ldots$?
  - answer: no. $(x = 2, y = 0)$
  - ...
- Most-helpful oracle
  - query: $\hat{h} = 111111\ldots$?
  - answer: no. $(x = 2020, y = 0)$
  - query: $\hat{h} = h_{999999}$?
  - answer: no. $(x = 2021, y = 1)$

Outline

Passive Learning (PAC Learning, Statistical Learning, Learning from iid Data)

Active Learning

Machine Teaching: Helpful Teachers

Online Learning

Multi-Armed Bandits

Reinforcement Learning

Teaching Protocol

$\mathcal{H}$ is common knowledge. Teacher has $h^* \in \mathcal{H}$ and knows the learner's algorithm

- Teacher creates teaching set $S = (x_1, y_1) \ldots (x_n, y_n) \in X \times Y$
- Learner receives S and selects $\hat{h} \in \mathcal{H}$
- Teacher's goals:
  - making the learner learn: $\hat{h} = h^*$
  - using the least effort: minimize n

Teaching Dimension

For learners that arbitrarily pick $\hat{h} \in V(S)$:

- S is a teaching set for h^* with respect to $\mathcal{H}$, if h^* is the only consistent hypothesis in $\mathcal{H}$.
- $TD(h^*, \mathcal{H}) =$
  the size of the smallest teaching set for h^* w.r.t. $\mathcal{H}$
- $TD(\mathcal{H}) = \max_{h \in \mathcal{H}} TD(h, \mathcal{H})$

Recall Example 1: $\mathcal{H} = \{h_i = 0 \ldots 0111111 \ldots : i \in \mathbb{N}\}$,
$h^* = h_{2021}$

- $S = \{(2020, 0), (2021, 1)\}$ is a teaching set
- ... so is $S = \{(2020, 0), (2021, 1), (2022, 1)\}$
- ... but not $S = \{(2020, 0)\}$ nor $S = \{(2020, 0), (2022, 1)\}$
- $TD(h_1, \mathcal{H}) = 1$; $TD(h_a, \mathcal{H}) = 1, \forall a \geq 2$
- ... and $TD(\mathcal{H}) = 2$

More Examples of Teaching Dimension

	x1 ... xn
h0	0000000000
h1	1000000000
h2	0100000000
h3	0010000000
	...
hn	0000000001

$$TD(\mathcal{H}) = n \gg VC(\mathcal{H}) = 1$$

More Examples of Teaching Dimension

	x1 ...	
h1	1000000000	00000
h2	0100000000	00001
h3	0010000000	00010
h4	0001000000	00011
	...	
h_{2^k}	0000000001	11111

$$TD(\mathcal{H}) = 1 \ll VC(\mathcal{H}) = k$$

Teaching as Coding

- message: target concept $h^* \in \mathcal{H}$
- language: S
- decoder: learning algorithm

A conceptual way to find S:

$$\begin{aligned} \min_S \quad & |S| \\ \text{s.t.} \quad & \hat{h}(S) = h^* \end{aligned}$$

or

$$\min_S \ \text{effort}(S) + \|\hat{h}(S) - h^*\|$$

Machine Teaching Summary

- Teaching set S forces learner to learn h^*
- Teaching Dimension $TD(h^*, \mathcal{H})$ lower-bounds all sample-based learning
- For example, on 1D threshold
  - passive learning requires $O(\frac{1}{\epsilon})$ samples
  - active learning requires $O(\log \frac{1}{\epsilon})$
  - teaching only requires 2 regardless of ϵ

Outline

Passive Learning (PAC Learning, Statistical Learning, Learning from iid Data)

Active Learning

Machine Teaching: Helpful Teachers

Online Learning

Multi-Armed Bandits

Reinforcement Learning

Online Learning Protocol

$\mathcal{H}$ is common knowledge. Environment has $h^* \in \mathcal{H}$

1. For $t = 1, 2, \ldots$
2. environment shows an arbitrary $x_t \in X$
    - no $P(x)$ assumption
3. learner predicts $\hat{y}_t$
4. environment reveals true label $h^*(x_t)$
5. learner updates model

Mistake Bound

Example $\mathcal{H} = \{h_i = 0 \ldots 0111111 \ldots : i \leq N\}$, $h^* = h_{2021}$

- If env keeps showing $x = 1$: no hope to learn h^*, but also no further mistakes
- Mistake bound on <u>any input sequence</u>
  - If env is a helpful teacher, mistake bound is $TD(\mathcal{H})$.
  - Assume worst case env instead

Some ERM Algorithms are No Good for Online Learning

- Trivial algorithm: Start with $V = \mathcal{H}$. Repeat:
  - Pick any $\hat{h} \in V$
  - Receive x_t, predict $\hat{h}(x_t)$, receive $h^*(x_t)$
  - $V \leftarrow \{h \in V : h(x_t) = h^*(x_t)\}$
- Trivial mistake bound: $|\mathcal{H}| - 1$
  - $h^* = h_1$, $\hat{h} = h_N$, $x = N - 1$; $\hat{h} = h_{N-1}$, $x = N - 2$; $\ldots$

The Halving Algorithm

- Start with $V = \mathcal{H}$. Repeat:
  - Receive x_t, predict majority vote by V, receive $h^*(x_t)$
  - $V \leftarrow \{h \in V : h(x_t) = h^*(x_t)\}$
- Any mistake cuts V by at least half
- Mistake bound $\log_2 |\mathcal{H}|$

Online Learning Summary

- No separate training/test, no iid data assumption
- Mistake bound, can generalize to regret (learning from experts)
- Halving is suboptimal: Littlestone dimension and Standard Optimal Algorithm

Outline

(Stochastic) Multi-Armed Bandit Protocol

1. Environment has k reward distributions $R_1, \ldots, R_k$ with mean $\mu_1, \ldots, \mu_k$
2. For $t = 1, 2, \ldots, T$
3. learner pulls arm $a_t \in \{1 \ldots k\}$
4. environment generates reward $r_t \sim R_{a_t}$

- Learner chooses which arm to pull, like in active learning
- Learner knows the R family (e.g. Bernoulli, Gaussian) but not the μ's
- Generalizes A/B testing

Example: $k = 2$ Bernoulli$\{0, 1\}$ Arms

- First pull $a_1 = 1$, $r_1 = 1$
- Second pull $a_2 = 2$, $r_2 = 0$
- Third pull?
- What if we have pulled arm1 10 times with $\hat{\mu}_1 = 0.7$, and arm2 5 times with $\hat{\mu}_2 = 0.4$?

Exploration Exploitation Tradeoff

Two distinct goals:

- Pure exploration = best arm identification

$$\max P\left(a_{T+1} \in \operatorname*{argmax}_a \mu_a\right)$$

- Regret minimization = maximizing cumulative reward $\sum_{t=1}^{T} r_t$

$$\text{Regret}(T) = \mu^* T - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$$

$$\mu^* = \max_a \mu_a$$

Upper Confidence Bound: Exploration Bonus

The UCB algorithm:

- For $t = 1, 2, \ldots, T$
- learner pulls arm

$$a_t \in \operatorname*{argmax}_{i \in [k]} \widehat{\mu}_i + \sqrt{\frac{4 \log T}{T_i}}$$

- receives r_t, updates $\widehat{\mu}_{a_t}, T_{a_t}$

Theorem

$$\mathrm{Regret}(T) \leq 8\sqrt{kT \log T} + 3 \sum_{i=1}^{k} (\mu^* - \mu_i).$$

"No regret" (per step, asymptotic)

With a Helpful Teacher

1. For $t = 1, 2, \ldots, T$
2. learner pulls arm $a_t \in \{1 \ldots k\}$
3. environment generates reward $r_t \sim R_{a_t}$
4. teacher modifies reward to $r_t + \delta_t$ before giving it to learner

- Guides best-arm identification
- Same vulnerability to adversarial attacks

Contextual Bandit

A context is a state $s \in S$

1. Environment has
   - context distribution ν
   - k reward distributions per state s: $R_{s1}, \ldots, R_{sk}$ with mean $\mu_{s1}, \ldots, \mu_{sk}$
2. For $t = 1, 2, \ldots, T$
3. environment shows state $s_t \sim \nu$
4. learner pulls arm $a_t \in \{1 \ldots k\}$
5. environment shows reward $r_t \sim R_{s_t, a_t}$

Useful if similar states share similar R's, e.g. linear bandits $\mu = \theta^\top \phi(s, a)$

Multi-Armed Bandit Summary

- Simplest exploration-exploitation tradeoff
- State-less (basic bandit) or memoryless (contextual bandit)

Outline

Passive Learning (PAC Learning, Statistical Learning, Learning from iid Data)

Active Learning

Machine Teaching: Helpful Teachers

Online Learning

Multi-Armed Bandits

Reinforcement Learning

Markov Decision Process

Contextual bandit + first-order state transition. Environment:

- State space S
- Action space A
- State transitions $P(s' \mid s, a)$
- Reward distributions $R(s, a)$
- Initial state distribution ν
- Discounting parameter $\gamma \in (0, 1)$

Reinforcement Learning Interaction Protocol

The learner's policies $\pi : S \mapsto$ probability simplex on A

1. Learner picks initial policy π_0
2. Environment draws initial state $s_0 \sim \nu$
3. For $t = 0, 1, 2, \ldots$
4. learner chooses (randomized) action $a_t \sim \pi_t(s_t)$
5. environment generates reward $r_t \sim R(s_t, a_t)$
6. environment transits learner to $s_{t+1} \sim P(\cdot \mid, s_t, a_t)$
7. learner updates policy π_{t+1}

Value Function, Optimal Policy, Regret

For a fixed π, define state-value function $V^{\pi} : S \mapsto \mathbb{R}$

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right]$$

Two distinct goals:

- Optimal policy identification

$$\pi^* \in \operatorname*{argmax}_{\pi} \mathbb{E}_{s\sim\nu} V^{\pi}(s)$$

- Regret minimization

$$\mathbb{E}\left[V^{\pi^*} - \sum_{t} \gamma^t r_t\right]$$

Solution Strategies

Three types of RL methods:

1. Model-based: estimate $\hat{P}, \hat{R}$ from experience, then plan in the estimated MDP
2. Value-based (e.g. Q-learning): estimate the optimal action-value Q^* function with value iteration (fixed point to Bellman optimality equations)

   $$Q(s,a) \leftarrow R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \max_{a'} Q(s', a')$$

   Then extract the optimal policy

   $$\pi^*(s) \in \operatorname*{argmax}_a Q(s,a)$$

3. Policy gradient (e.g. REINFORCE): parametrize π_θ, then directly optimize

   $$\max_\theta \mathbb{E}_{s \sim \nu} V^{\pi_\theta}(s)$$

Upper Confidence Bound Value Iteration (UCBVI)

Episodic MDP with horizon H. Assume reward function R known.

1. For episode $k = 0, \ldots, K-1$
2. Form empirical transition estimate $\hat{P}_h^k$
3. Form reward bonus $b_h^k(s,a) = H\sqrt{\frac{\log \frac{SAHK}{\delta}}{T_h^k(s,a)}}$
4. $\pi^k = \text{ValueIteration}(\hat{P}^k, R + b_h^k : h = 0 \ldots H-1)$
5. Run π^k to generate a new trajectory, add to data

Theorem

Regret bound of UCBVI

$$\text{Regret} = \mathbb{E}\left[\sum_{k=0}^{K-1}\left(V^* - V^{\pi^k}\right)\right] \leq 2H^2 S\sqrt{AK\log(SAH^2K^2)}$$

RL With a Helpful Teacher 1

Imitation learning

- Expert provides trajectories

  $$(s_0, a_0, s_1, a_1, \ldots)$$

  but no reward r_t is observed.
- Goal: learn $\hat{\pi}$ as good as the expert
  - Require specialized learner (not standard RL)
  - Behavior cloning: reduction to supervised learning $\pi : S \mapsto A$
  - Inverse reinforcement learning: estimate reward function $R(s, a)$, then planning

RL With a Helpful Teacher 2

- Teacher shaping the interaction trajectories

  $$\text{on rewards: } (s_0, a_0, r_0 + \delta_0, s_1, a_1, r_1 + \delta_1, \ldots)$$

  $$\text{on transitions: } (s_0, a_0, r_0, s'_1, a_1, r_1, s'_2, \ldots)$$

  or both.
- Standard RL learner
- Goal: guide the learner to π^* faster
- Teacher planning for δ_t or s'_t: a higher-level RL problem; state includes learner $\hat{\pi}_t$

References

Passive learning, online learning

- Understanding Machine Learning: From Theory to Algorithms. Shalev-Shwartz and Ben-David, 2014

Active learning

- Theory of Active Learning. Hanneke, 2014

Machine teaching

- An Overview of Machine Teaching. Zhu, Singla, Zilles, and Rafferty. 2018

Multi-Armed Bandits

- Bandit Algorithms. Lattimore and Szepesvari. 2020

Reinforcement Learning

- Reinforcement Learning: Theory and Algorithms. Agarwal, Jiang, Kakade, Sun. (draft 2021)