Foundations of Reinforcement Learning

Imitation learning

Yuejie Chi

Department of Electrical and Computer Engineering

Carnegie Mellon University

Spring 2023

Outline

The pure offline setting: behavior cloning

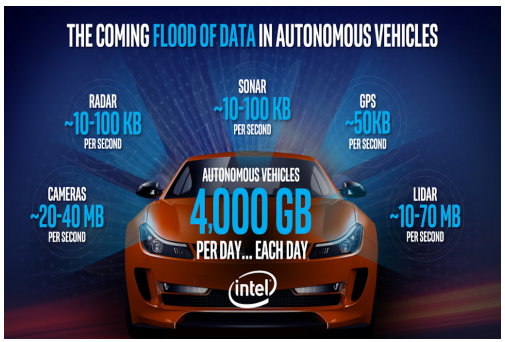
The hybrid setting: MaxEnt IRL

The interactive setting: DAgger

Offline RL / Batch RL

- Sometimes we can not explore or generate new data
- But we have already stored tons of historical data

medical records

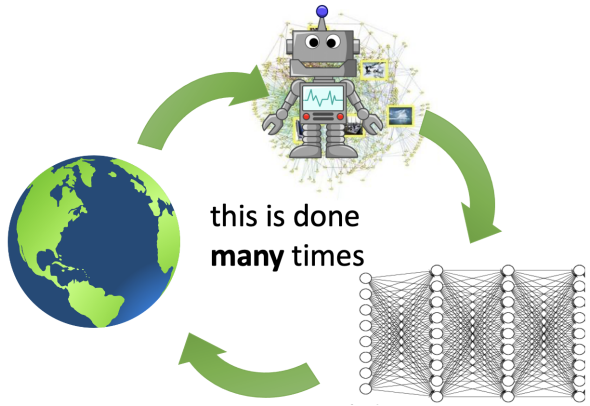


data of self-driving

clicking times of ads

Can we learn a good policy based solely on historical data without active exploration?

Online versus Offline RL

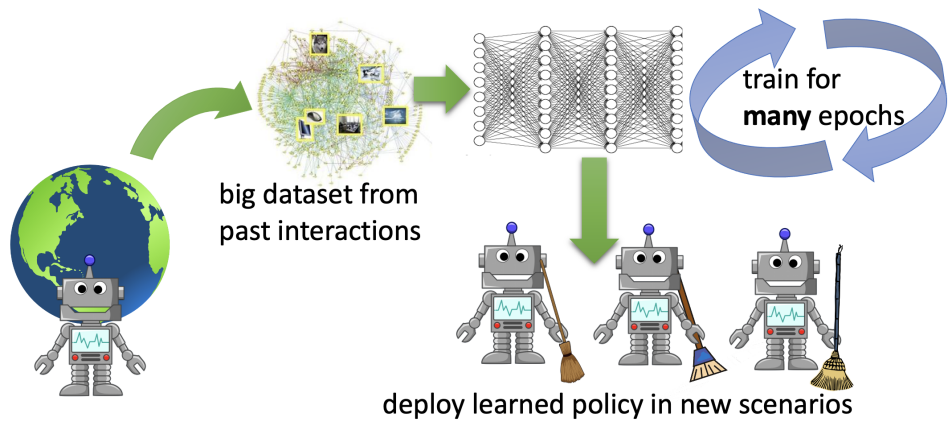


Online RL

- interact with environment
- actively collect new data

Offline/Batch RL

- no interaction
- data is given



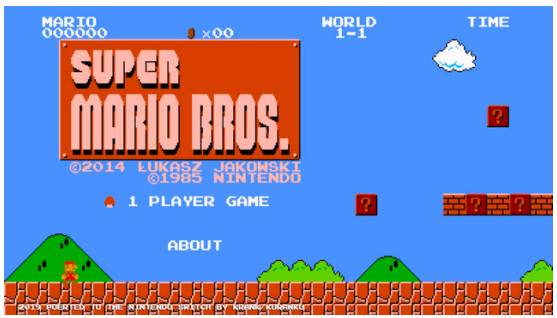
Two main types of offline data and approaches

- Expert data: e.g., expert demonstration
  - imitation learning (imitate experts' behavior)
- Uniform coverage data: e.g., generative model / simulator
  - a different set of algorithms (e.g., model-based, model-free methods)

Imitation learning (IL)

Imitation is the sincerest form of flattery that mediocrity can pay to greatness.

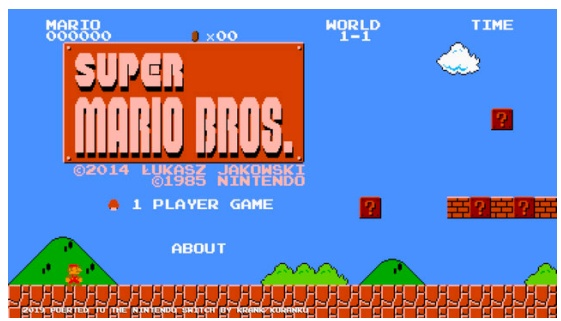
— *Oscar Wilde*



Goal: learn a good policy that mimics the expert demonstration.

Behavior cloning: IL from offline data

- Reward function is unknown or hard-to-tune in practice.
- Behavior cloning leverages expert demonstration to directly learn a policy without inferring the reward function.



Many successes: robots, autonomous driving, drones, ...

Three settings

Pure offline setting (behavior cloning):

- Only expert demonstration is available

Hybrid setting (MaxEnt IRL):

- Expert demonstration is available
- Able to interact with the real world (e.g. through the ground truth transition dynamic)

Interactive setting (DAgger):

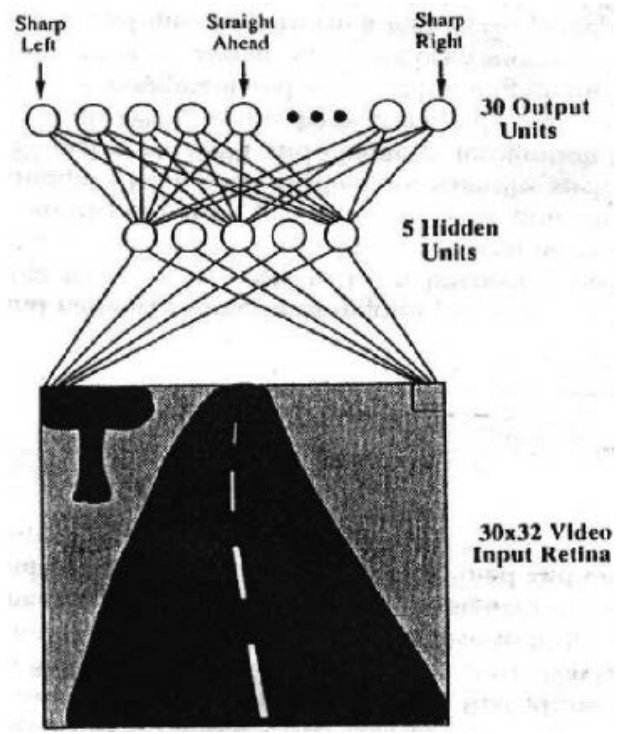
- Access to an interactive expert
- Able to interact with the real world

The pure offline setting: behavior cloning

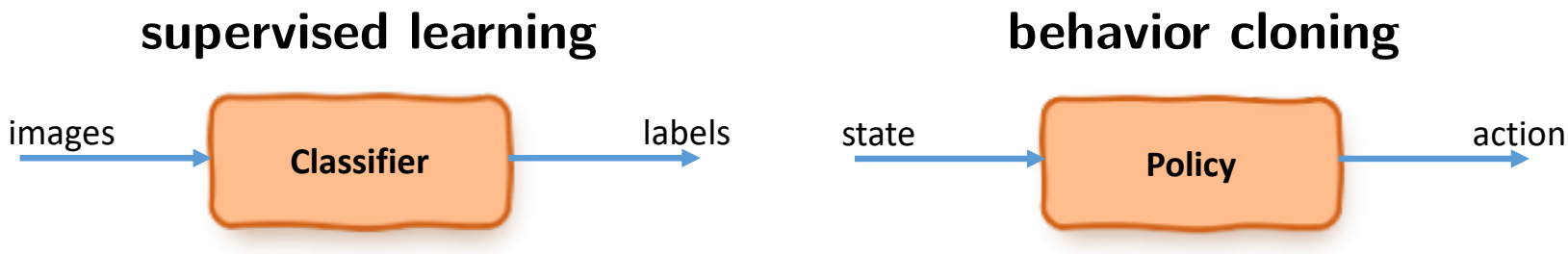
ALVINN

ALVINN (Autonomous Land Vehicle In a Neural Network), the self-driving car from 1988 [Pomerleau, 1988]!

Photo 1: *The NAVLAB autonomous navigation test-bed vehicle and the road used for trial runs.*



Reduction to supervised learning



Invoke supervised learning approaches to learn the policy:
a state-to-action mapping!

- Break down the expert demonstration (e.g., a trajectory) into a training dataset of (state, action) pairs;
- Learn a state-to-action mapping (i.e., policy) from the training data via your favorite supervised learning algorithm.

— *doesn't require knowing the reward!*

Supervised learning

Expert data: suppose we have an i.i.d. dataset

$$\mathcal{D} = \{s_i, a_i\}_{i=1}^N, \qquad \text{with} \quad a_i = \pi^\star(s_i), \;\; s_i \sim d_\rho^{\pi^\star}.$$

Here, $\pi^\star$ is the optimal policy, and $d_\rho^{\pi^\star}$ is the discounted state-visitation distribution induced by $\pi^\star$.

Supervised learning: we learn a policy

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^N \ell(\pi; s_i, a_i)$$

- Π is the policy class, which is assumed to be finite, and $\pi^\star \in \Pi$
- $\ell(\pi; s_i, a_i)$ is the sample loss, e.g.
  - negative log-likelihood: $-\log \pi(a_i|s_i)$
  - least-squares loss: $\|\pi(s_i) - a_i\|_2^2$
  - hinge loss, 0-1 loss, etc...

Performance guarantee of behavior cloning

Theorem 1 ([Ross and Bagnell, 2010])

Suppose that supervised learning works, where the learned deterministic policy satisfy

$$\mathbb{E}_{s\sim d_\rho^{\pi^\star}} \mathbb{I}\left(\widehat{\pi}(s) \neq \pi^\star(s)\right) \leq \epsilon,$$

then BC returns a policy $\widehat{\pi}$ such that

$$V^\star(\rho) - V^{\hat{\pi}}(\rho) \leq \frac{2}{(1-\gamma)^2}\epsilon.$$

- The error is "amplified" by a quadratic factor of the horizon dependency $\frac{1}{(1-\gamma)^2}$, which is unavoidable in the worst case [Rajaraman et al., 2020].

Proof of Theorem 2

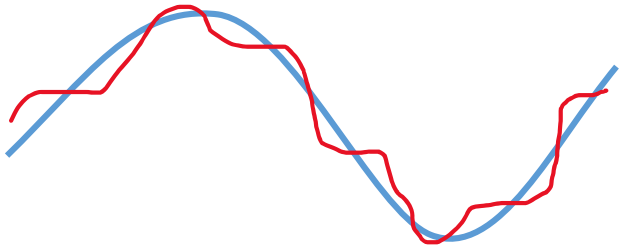
By the performance difference lemma,

$$\begin{aligned}
V^{\pi^\star}(\rho) - V^{\hat{\pi}}(\rho) &= \frac{1}{(1-\gamma)} \mathbb{E}_{s \sim d_\rho^\star} A^{\hat{\pi}}(s, \pi^\star(s)) \\
&= \frac{1}{(1-\gamma)} \Big[\mathbb{E}_{s \sim d_\rho^\star} A^{\hat{\pi}}(s, \pi^\star(s)) - \underbrace{\mathbb{E}_{s \sim d_\rho^\star} A^{\hat{\pi}}(s, \widehat{\pi}(s))}_{=:0} \Big] \\
&\leq \frac{2}{(1-\gamma)^2} \mathbb{E}_{s \sim d_\rho^\star} \mathbb{I}\left(\widehat{\pi}(s) \neq \pi^\star(s)\right) \\
&\leq \frac{2\epsilon}{(1-\gamma)^2},
\end{aligned}$$

where the penultimate line used the fact that $-\frac{1}{1-\gamma} \leq A^{\hat{\pi}} \leq \frac{1}{1-\gamma}$, and the last line used $\mathbb{E}_{s \sim d_\rho^{\pi^\star}} \mathbb{I}\left(\widehat{\pi}(s) \neq \pi^\star(s)\right) \leq \epsilon$.

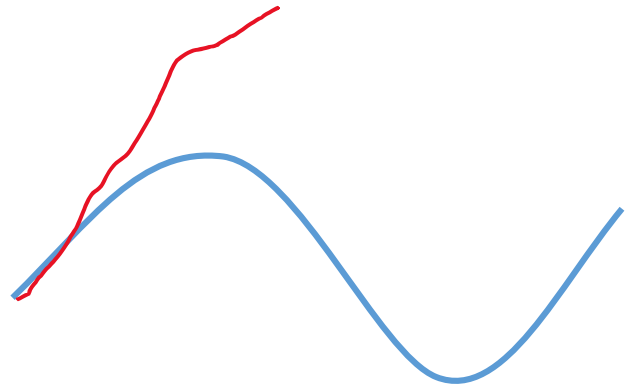
Why does error compounding occur?

Consider prediction over a horizon H: error at step t is ϵ.



supervised learning
independent error over t

$$\epsilon + \ldots + \epsilon = H\epsilon$$



behavior cloning
error propagates into the future

$$H\epsilon + (H-1)\epsilon + \ldots + \epsilon \sim H^2\epsilon$$

The BC error is quadratic in H instead of linear in H!

Error rate of supervised learning?

Taking Maximum likelihood estimate (MLE) as an example...

$$\widehat{\pi}_{\mathsf{MLE}} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{N} \log \pi(a_i|s_i)$$

Theorem 2 ([Agarwal et al., 2019])

With probability at least $1-\delta$*, we have*

$$\mathbb{E}_{s \sim d_\rho^{\pi^\star}} \left\| \widehat{\pi}(\cdot|s) - \pi^\star(\cdot|s) \right\|_{\mathsf{TV}}^2 \leq \frac{2\log(|\Pi|/\delta)}{N}$$

- The error depends on $\log|\Pi|$, allowing rich policy class.

Performance guarantees of BC with MLE

Theorem 3 ([Agarwal et al., 2019])

With probability at least $1-\delta$*, BC returns a policy* $\widehat{\pi}_{\mathsf{MLE}}$ *such that*

$$V^{\star}(\rho) - V^{\hat{\pi}_{\mathsf{MLE}}}(\rho) \leq \frac{3}{(1-\gamma)^2}\sqrt{\frac{\log(|\Pi|/\delta)}{N}}.$$

- To achieve ϵ-accuracy, the sample size needs to be at least

$$N \gtrsim \frac{\log|\Pi|}{(1-\gamma)^4\epsilon^2}.$$

Proof of Theorem 3

Denote $\hat{\pi} := \hat{\pi}_{\mathsf{MLE}}$ for short. By the performance difference lemma,

$$
\begin{aligned}
V^{\pi^\star}(\rho) - V^{\hat{\pi}}(\rho) &= \frac{1}{(1-\gamma)} \mathbb{E}_{s\sim d_\rho^\star} A^{\hat{\pi}}(s,a) \\
&= \frac{1}{(1-\gamma)} \Big[\mathbb{E}_{s\sim d_\rho^\star} \mathbb{E}_{a\sim\pi^\star(\cdot|s)} A^{\hat{\pi}}(s,a) - \underbrace{\mathbb{E}_{s\sim d_\rho^\star} \mathbb{E}_{a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a)}_{=:0} \Big] \\
&\le \frac{1}{(1-\gamma)^2} \mathbb{E}_{s\sim d_\rho^\star} \left\| \widehat{\pi}(\cdot|s) - \pi^\star(\cdot|s) \right\|_1 \\
&\le \frac{1}{(1-\gamma)^2} \sqrt{\mathbb{E}_{s\sim d_\rho^\star} \left\| \widehat{\pi}(\cdot|s) - \pi^\star(\cdot|s) \right\|_1^2} \\
&= \frac{1}{(1-\gamma)^2} \sqrt{4\mathbb{E}_{s\sim d_\rho^\star} \left\| \widehat{\pi}(\cdot|s) - \pi^\star(\cdot|s) \right\|_{\mathsf{TV}}^2}
\end{aligned}
$$

where we used $(\mathbb{E}[X])^2 \le \mathbb{E}[X^2]$.

Distribution shift

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim d^{\pi^\star}} \left[\ell(\pi; s, \pi^\star(s)) \right]$$

In supervised learning, a learner's prediction does not influence the distribution of examples upon which it will be tested.

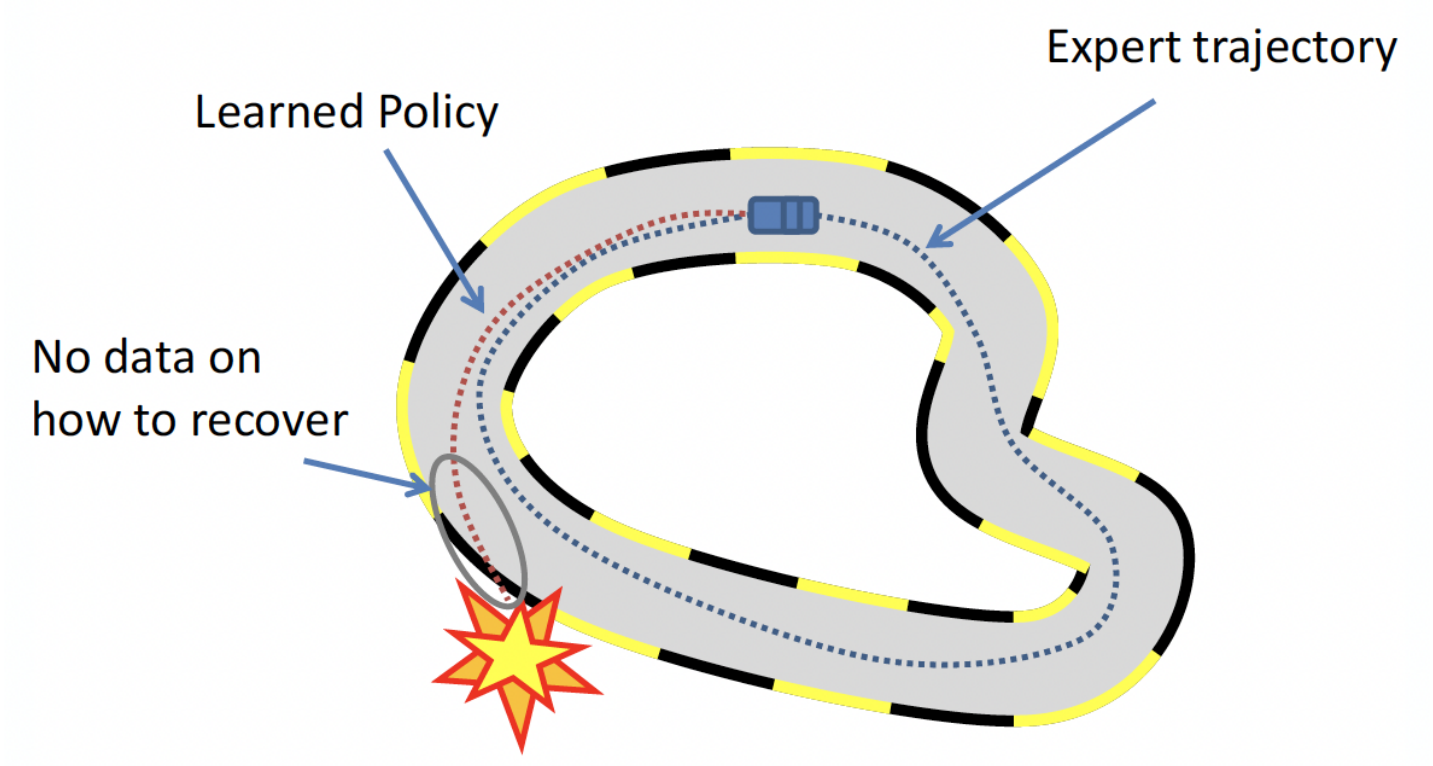
training: $s \sim d^{\pi^\star}$ $\quad = \quad$ test: $s \sim d^{\pi^\star}$

This is not the case with sequential decision making: Predictions affect future inputs/ observations!

training: $s \sim d^{\pi^\star}$ $\quad \neq \quad$ test: $s \sim d^{\hat{\pi}}$

Distribution shift

training: $s \sim d^{\pi^\star}$ $\neq$ test: $s \sim d^{\hat{\pi}}$



ALVINN [Pomerleau, 1988]: "when driving for itself, the network may occasionally stray from the road center, so it must be prepared to recover by steering the vehicle back to the center of the road."

The hybrid setting: Maximum Entropy Inverse RL

The hybrid setting

Expert data: suppose we have an i.i.d. dataset

$$\mathcal{D} = \{s_i, a_i\}_{i=1}^N, \qquad \text{with} \quad a_i = \pi^\star(s_i), \;\; s_i \sim d_\rho^{\pi^\star}.$$

Here, $\pi^\star$ is the optimal policy, and $d_\rho^{\pi^\star}$ is the discounted state-visitation distribution induced by $\pi^\star$.

Additionally, we known the transition kernel $P(\cdot|s,a)$ of the underlying MDP

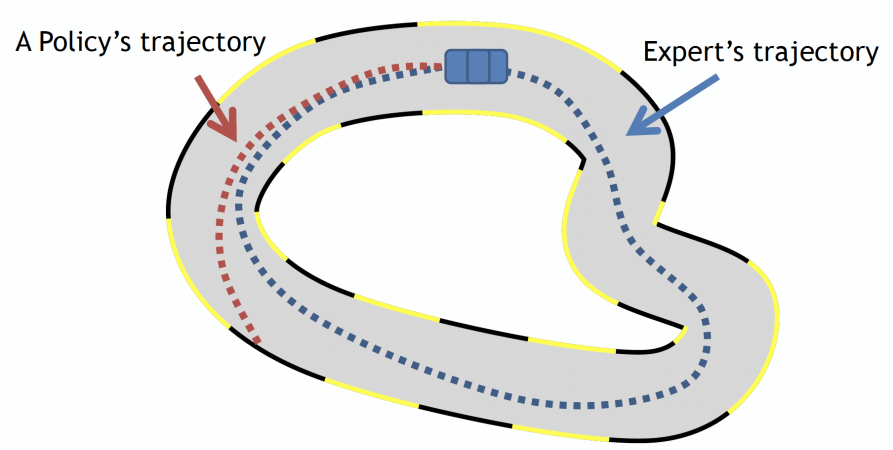
- Enables planning once we are given a reward function.

What is the benefit of this additional piece of information?

Intuition and distribution matching

Intuition: by rolling out a policy π in the environment, we can detect its deviation from the expert policy $\pi^\star$ through **distribution matching**:

$$d^\pi \approx d^{\pi^\star}?$$



Theorem 4 ([Agarwal et al., 2019])

There exists a computationally-inefficient algorithm (distribution matching) such that with probability at least $1-\delta$, it returns a policy $\widehat{\pi}_{\mathsf{DM}}$ such that

$$V^\star(\rho) - V^{\hat{\pi}_{\mathsf{DM}}}(\rho) \lesssim \frac{1}{(1-\gamma)}\sqrt{\frac{\log(|\Pi|/\delta)}{N}}.$$

Maximum Entropy Inverse RL (MaxEnt IRL)

MaxEnt IRL: A popular computationally-efficient approach in practice [Ziebart et al., 2008].

- Denote ρ^π the trajectory distribution induced by π:

$$\rho^\pi = \rho(s_0)\pi(a_0|s_0)P(s_1|s_0,a_0)\pi(a_1|s_1)\dots$$

- Maximize the entropy of the trajectory while matching with the expert:

$$\underbrace{\max_\pi \mathcal{H}(\rho^\pi)}_{\text{entropy maximization}} = \max_\pi \mathbb{E}_{s,a\sim d^\pi} -\log \pi(a|s)$$

$$\text{s.t.} \quad \underbrace{\mathbb{E}_{s,a\sim d^\pi}\phi(s,a) = \mathbb{E}_{s,a\sim d^{\pi^\star}}\phi(s,a)}_{\text{distribution matching}}$$

where $\phi(s,a) \in \mathbb{R}^d$ is the feature map for state-action pair (s,a).

MaxEnt IRL: algorithm and interpretation

- Using Lagrange formulation:

$$\min_{\pi} \mathbb{E}_{s,a\sim d^{\pi}} \log \pi(a|s) + \max_{\theta} \left(\mathbb{E}_{s,a\sim d^{\pi}} \theta^{\top}\phi(s,a) - \mathbb{E}_{s,a\sim d^{\pi^{\star}}} \theta^{\top}\phi(s,a)\right),$$

  where θ is the Lagrangian multiplier.

- By the von Neumann's minimax theorem:

$$\max_{\theta} \min_{\pi} \left[\mathbb{E}_{s,a\sim d^{\pi}} \log \pi(a|s) + \mathbb{E}_{s,a\sim d^{\pi}} \theta^{\top}\phi(s,a) - \mathbb{E}_{s,a\sim d^{\pi^{\star}}} \theta^{\top}\phi(s,a)\right]$$

- Saddle-point optimization: alternatively update θ (via gradient ascent) and π (via planning).

MaxEnt IRL: algorithm and interpretation

For $t = 0, 1, \ldots$

$$\pi_t = \operatorname*{argmin}_{\pi} \; \mathbb{E}_{s,a \sim d^{\pi}} \underbrace{\left(\log \pi(a|s) + \theta_t^{\top} \phi(s,a)\right)}_{\text{entropy-regularized cost}}$$

$$\theta_{t+1} = \theta_t + \eta \left(\mathbb{E}_{s,a \sim d^{\pi_t}} \phi(s,a) - \mathbb{E}_{s,a \sim d^{\pi^\star}} \phi(s,a)\right)$$

Interpreting $r_\theta(s,a) \approx \theta^{\top} \phi(s,a)$,

- Optimizing π amounts to planning in an entropy-regularized MDP, which can be found by **soft value iteration**.
- Optimizing θ amounts to learning the cost function (or reward function).

The interactive setting: DAgger

Revisit the supervised learning reduction

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim d^{\pi}} \left[\ell(\pi; s, \pi^{\star}(s)) \right]$$

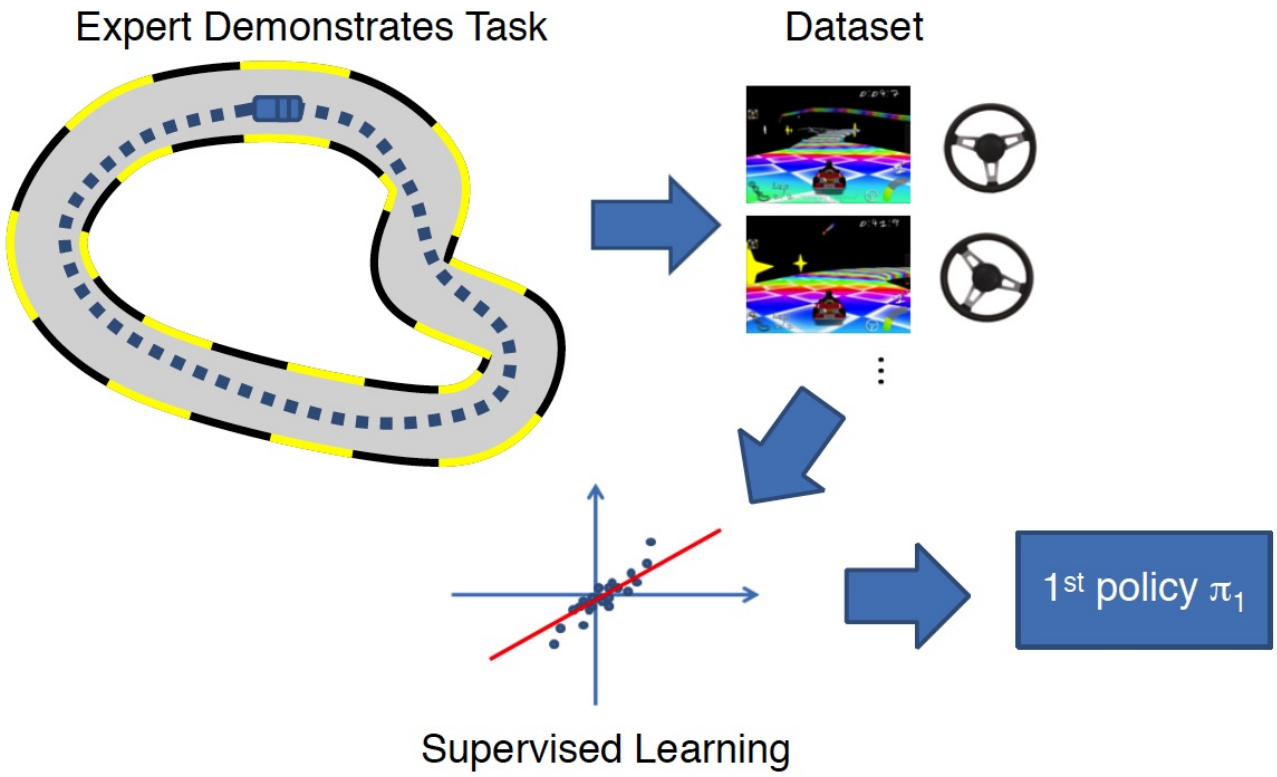
We want to optimize the performance when the state is drawn from d^{π}, which also need to be learned.

- non-i.i.d. supervised learning problem
- much more challenging!

[Ross et al., 2011] proposed an algorithm called `DAgger` (dataset aggregation) that allows sampling new expert data on the learned policy.

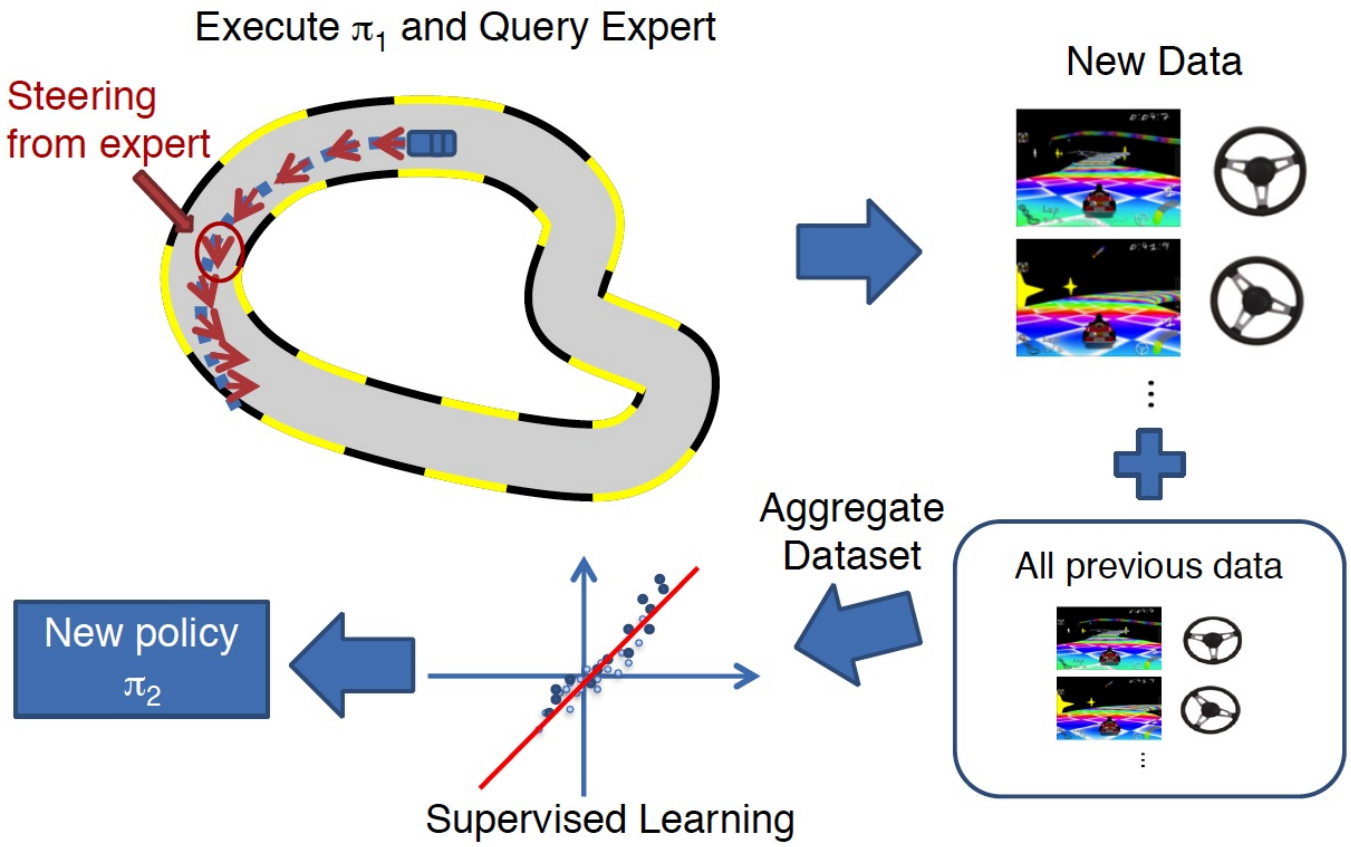
DAgger: dataset aggregation

[Ross et al., 2011]: train on a mixture of expert and behavior policy via reduction to no-regret online learning



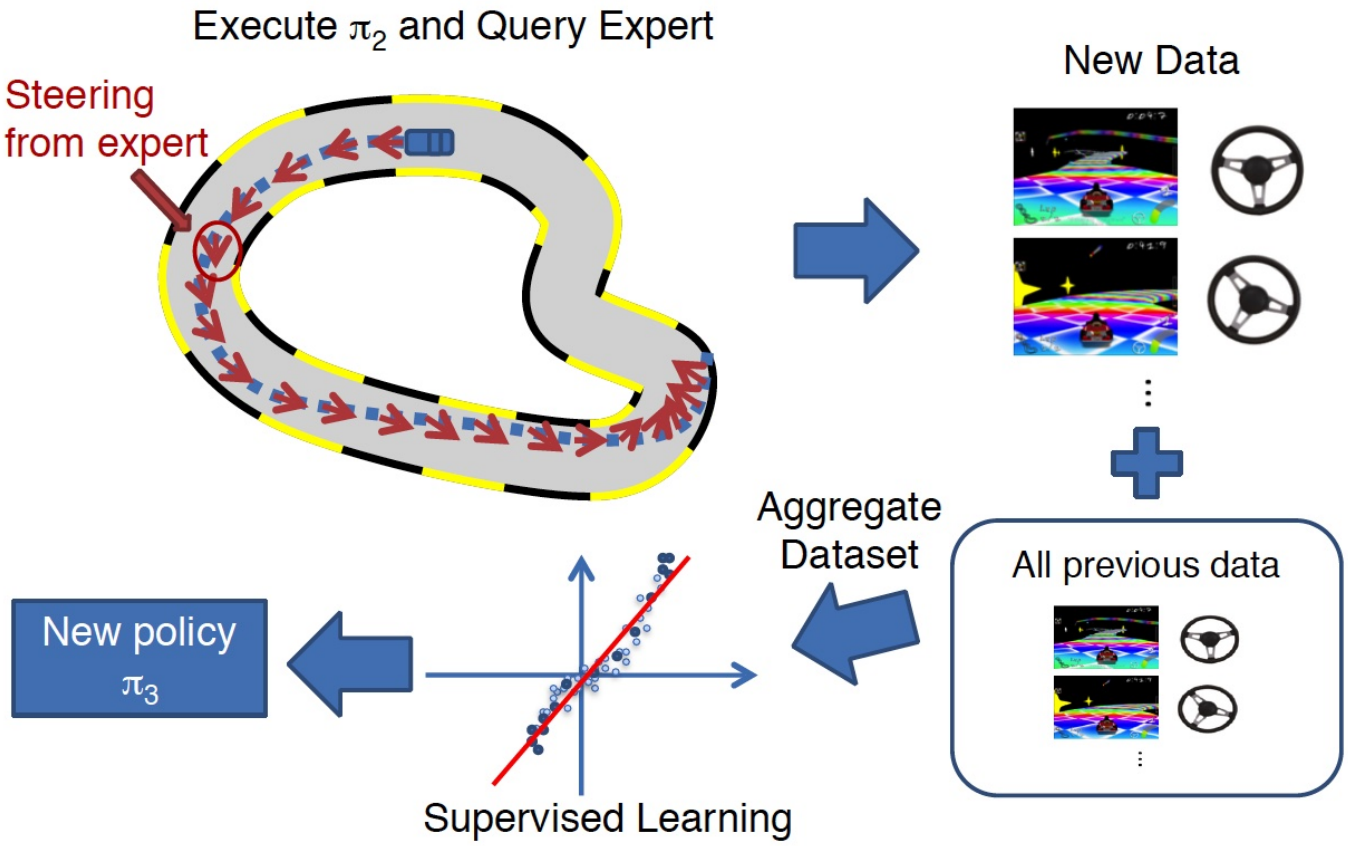
DAgger: dataset aggregation

[Ross et al., 2011]: train on a mixture of expert and behavior policy via reduction to no-regret online learning



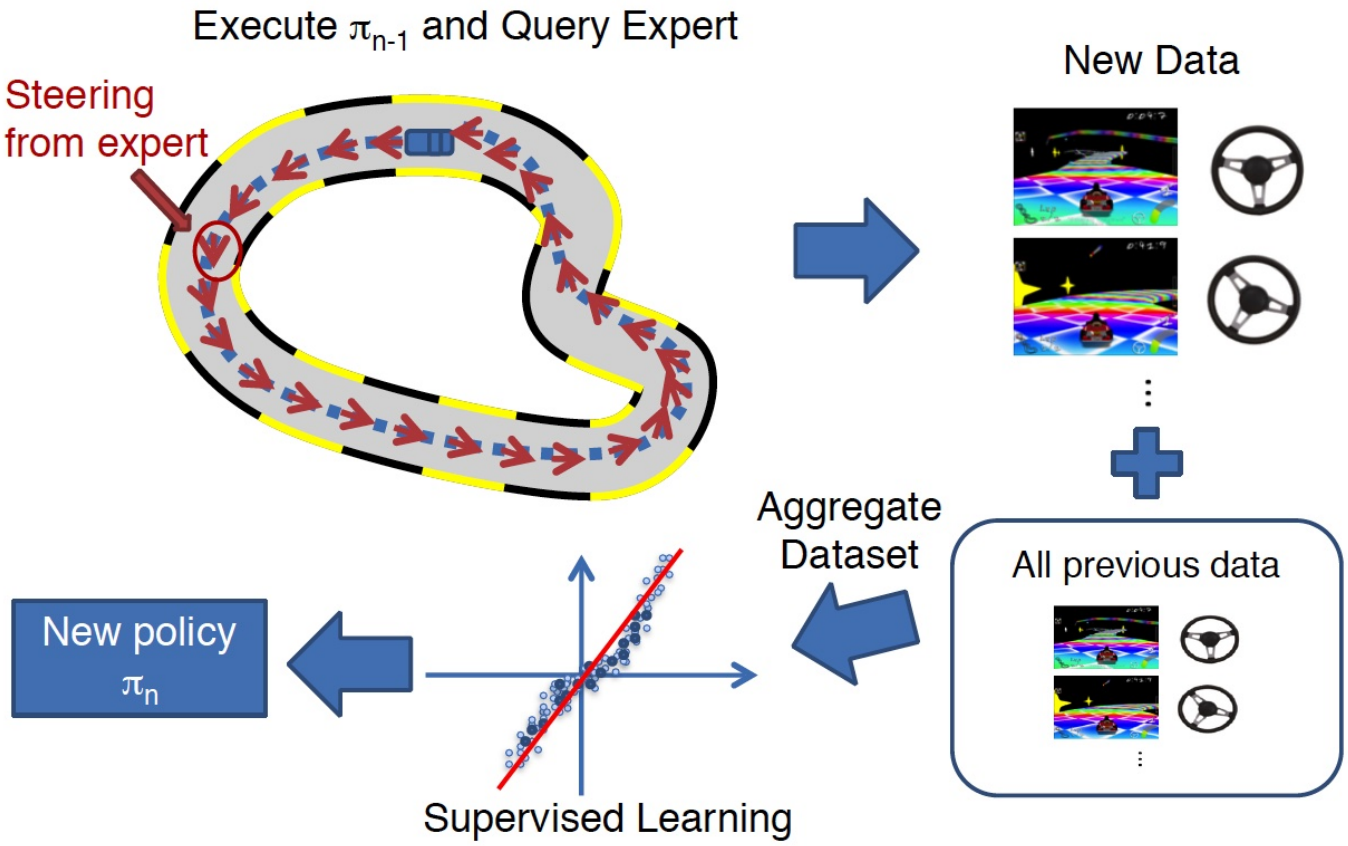
DAgger: dataset aggregation

[Ross et al., 2011]: train on a mixture of expert and behavior policy via reduction to no-regret online learning



DAgger: dataset aggregation

[Ross et al., 2011]: train on a mixture of expert and behavior policy via reduction to no-regret online learning



References I


Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019).
Reinforcement learning: Theory and algorithms.


Pomerleau, D. A. (1988).
ALVINN: An autonomous land vehicle in a neural network.
Advances in neural information processing systems, 1.


Rajaraman, N., Yang, L., Jiao, J., and Ramchandran, K. (2020).
Toward the fundamental limits of imitation learning.
Advances in Neural Information Processing Systems, 33:2914–2924.


Ross, S. and Bagnell, D. (2010).
Efficient reductions for imitation learning.
In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings.


Ross, S., Gordon, G., and Bagnell, D. (2011).
A reduction of imitation learning and structured prediction to no-regret online learning.
In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.


Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008).
Maximum entropy inverse reinforcement learning.
In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA.