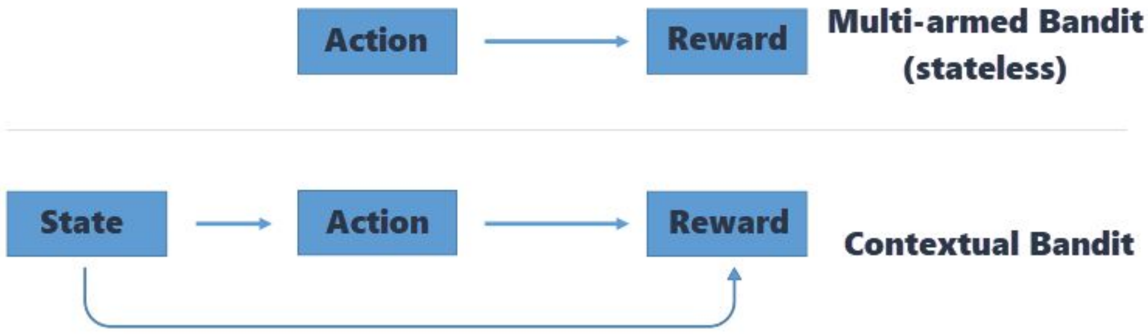
Imitation Learning

Roberto Capobianco

SAPIENZA
UNIVERSITÀ DI ROMA

Recap

From Multi-Armed to Contextual Bandits



Contextual bandits add back some context (state)

Contextual Bandits: Interaction

The interactive process that we deal with in CB is the following:

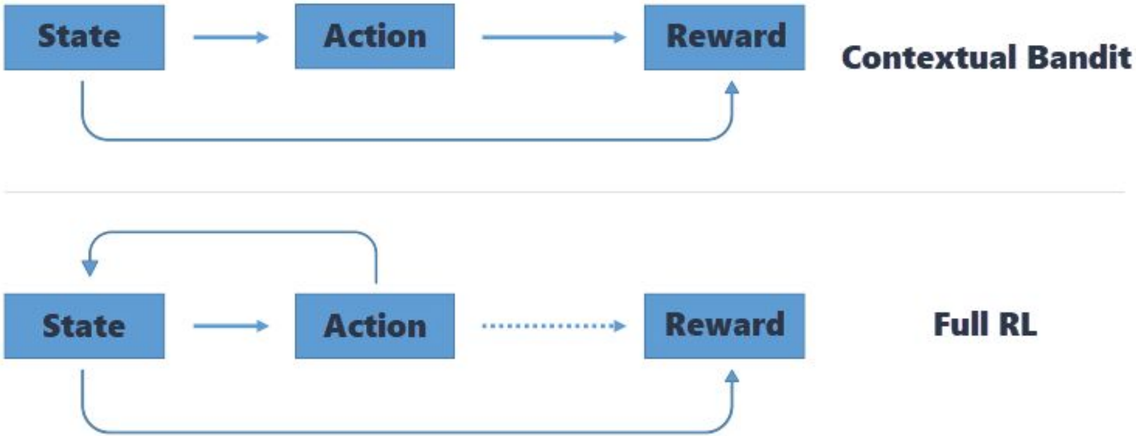
For t = 0, ..., T-1:

1. A new i.i.d. context x_t in X appears
2. Select an action a_t in A based on historical information and context
3. Observe reward $r(x_t,a_t)$ (which is context and arm dependent)

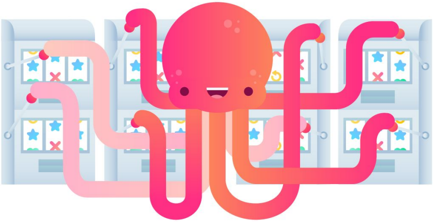
For simplicity we assume deterministic rewards, as the context is the challenge here

Contextual Bandits VS RL

In RL, conversely, states depend on previous actions: **we can say that contextual bandits are Finite-Horizon MDPs with horizon 1**



Contextual Bandits: Regret



Optimal policy: $\pi^\star = \arg\max_{\pi \in \Pi} \mathbb{E}_{x \sim \mu} r(x, \pi(x))$

At every iteration $a_t = \pi_t(x_t)$ is selected and a reward $r(x_t, a_t)$ is received: the regret is the **total expected reward if we always use π^*** VS the **total expected reward if we use our learned sequence of policies**

$$\text{Regret}_T = \boxed{T\mathbb{E}_{x \sim \mu}[r(x, \pi^\star(x))]} - \boxed{\sum_{t=0}^{T-1} \mathbb{E}_{x \sim \mu}[r(x, \pi^t(x))]}$$

Note that policies are different at every iteration t

Explore & Commit Algorithm

1. For t = 0, ..., N-1: **(explore)**
   - observe state $x_t \sim \mu$
   - uniform-randomly sample $a_t \sim \text{Unif}(A)$
   - observe reward $r_t = r(x_t, a_t)$
   - build, for x_t, an unbiased estimate of $\boxed{\mathbb{E}_{a_t \sim p}\hat{\mathbf{r}}[a] = r(x_t, a), \forall a}$
2. Compute policy

$$\hat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=0}^{N-1} \hat{\mathbf{r}}_i[\pi(x_i)]$$

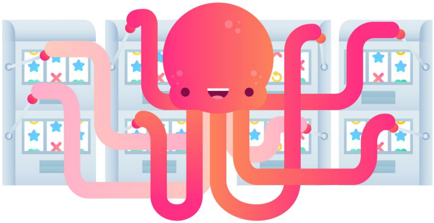
Given we are sampling from Unif(A)

$$\hat{\mathbf{r}}_t[a] = \begin{cases} 0 & a \neq a_t \\ \frac{r_t}{1/|\mathcal{A}|} & a = a_t \end{cases}$$

3. For t = N, ..., T-1: **(commit)**
   - observe state $x_t \sim \mu$
   - play arm

$$\text{Regret}_T = T\mathbb{E}_{x\sim\mu}[r(x, \pi^\star(x))] - \sum_{t=0}^{T-1} \mathbb{E}_{x\sim\mu}[r(x, \pi^t(x))] = O\left(T^{2/3}K^{1/3} \cdot \ln(|\Pi|)^{1/3}\right)$$

ε-Greedy



Instead of setting a threshold for exploring and then committing, we can try to interleave exploration and exploitation

1. For t = 0, ..., T: **(interleave exploration & exploitation)**
   - observe state $x_t \sim \mu$
   - $a_t \sim p_t = (1-\varepsilon)\delta(\pi^t(x_t)) + \varepsilon \mathrm{Unif}(A)$
   - observe reward $r_t = r(x_t, a_t)$
   - build, for x_t, an unbiased estimate of $\mathbb{E}_{a_t \sim p}\hat{\mathbf{r}}[a] = r(x_t, a), \forall a$
2. Update policy

$$\pi^{t+1} = \arg\max_{\pi \in \Pi} \sum_{i=0}^{t} \hat{\mathbf{r}}_i[\pi(x_i)]$$

ε = 0 -> exploit

ε = 1 -> uniformly explore

Bayesian Bandits

So far we have made no assumptions about the reward distribution ν_i, we only derived bounds on rewards

In Bayesian Bandits, however:

- We exploit *prior* knowledge of rewards
- Update a *posterior distribution* of rewards based on historical information
- Use posterior to guide exploration using:
  - upper confidence bounds (Bayesian UCB)
  - probability matching (Thompson Sampling)

Gaussian Bayesian Bandits: UCB

Now we are modelling a distribution, so we already have confidence

What is confidence for Gaussians? **standard deviation**

Let's do UCB by selecting the action with highest standard deviation

$$a_t = \text{argmax}_{i \text{ in } K}\, \mu_t(i) + c\sigma_t(i)/\sqrt{N_t(i)}$$

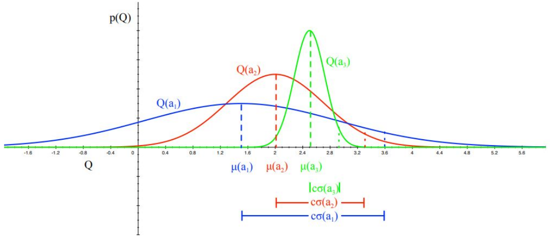
Gaussian Bayesian Bandits: Thompson Sampling

This is an estimation of the reward, in more generic MDPs this can be replaced with the Q function: we estimate a distribution of Q

For t = 0, ..., T:

1. for each arm i = 1, ..., K:
    - sample $\boxed{\hat{\mathbf{r}}_i}$ independently from $N(\mu_{t-1}(i), \sigma^2_{t-1}(i))$
2. pull arm

$$I_t = \arg\max_{i \in [K]} \hat{\mathbf{r}}_i$$



3. observe reward r_t
4. update posterior distribution $p(\mu_t(i), \sigma^2_t(i) | r_t)$

This can be done with different distributions as well

End Recap

Imitation Learning

Given an MDP, what happens if we cannot get access to a reward?

Imitation Learning

Given an MDP, what happens if we cannot get access to a reward?

We can learn by imitation of an expert!

Imitation Learning

Given an MDP, what happens if we cannot get access to a reward?

We can learn by imitation of an expert!

1. Collect expert demonstrations

$$D = \{s_i^*, a_i^*\}_{i=1}^M \sim d^{\pi^*}$$

For simplicity, let's assume expert is a (nearly) optimal policy π^*

Imitation Learning

Given an MDP, what happens if we cannot get access to a reward?

We can learn by imitation of an expert!

1. Collect expert demonstrations
2. Use a machine learning algorithm to learn to map states to actions

i.e., do regression or classification

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \ell\left(\pi, s^\star, a^\star\right)$$

loss can be negative likelihood

$$-\ln \pi(a^\star \mid s^\star)$$

or square error

$$\|\pi(s) - a^\star\|_2^2$$

Imitation Learning

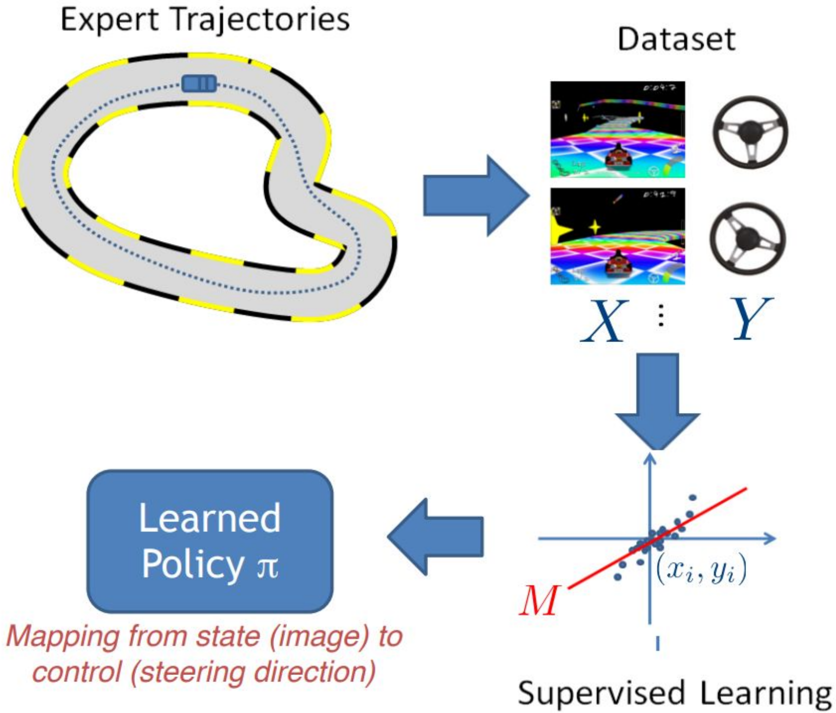
Given an MDP, what happens if we cannot get access to a reward?

We can learn by imitation of an expert!

1. Collect expert demonstrations
2. Use a machine learning algorithm to learn to map states to actions
3. Generate a policy

For simplicity, let's assume expert is a (nearly) optimal policy π^*

Behavior Cloning



Credits: Wen Sun

Behavior Cloning

Behavior Cloning: Issues

Behavior cloning, with probability 1-δ, returns a policy such that

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

(you can prove it using performance difference lemma)

Behavior Cloning: Issues

Behavior cloning, with probability 1-δ, returns a policy such that

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

Quadratic!

(you can prove it using performance difference lemma)

Behavior Cloning: Issues

Behavior cloning, with probability 1-δ, returns a policy such that

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

Quadratic!

(you can prove it using performance difference lemma)

Why?

Behavior Cloning: Issues

Behavior cloning, with probability 1-δ, returns a policy such that

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

Quadratic!

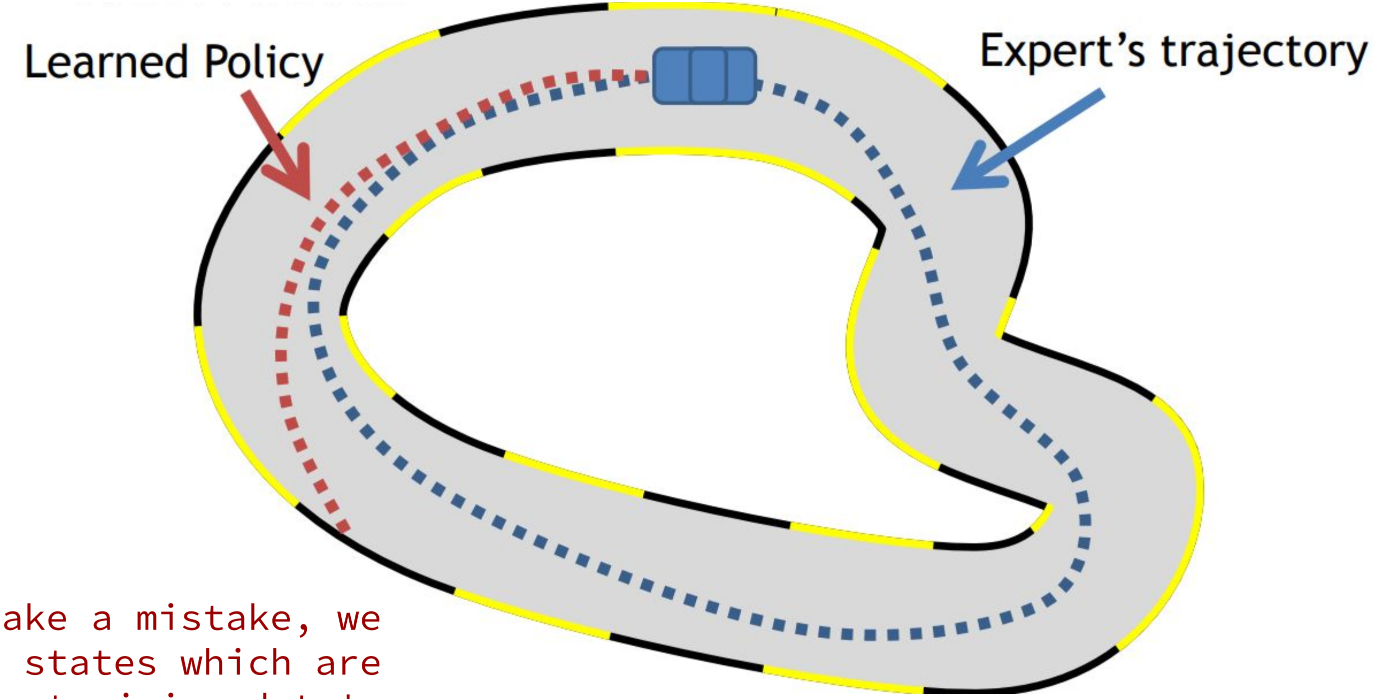
(you can prove it using performance difference lemma)

Why? Predictions affect future inputs/observations, inducing a distribution shift

SAPIENZA
UNIVERSITÀ DI ROMA

Credits: Wen Sun

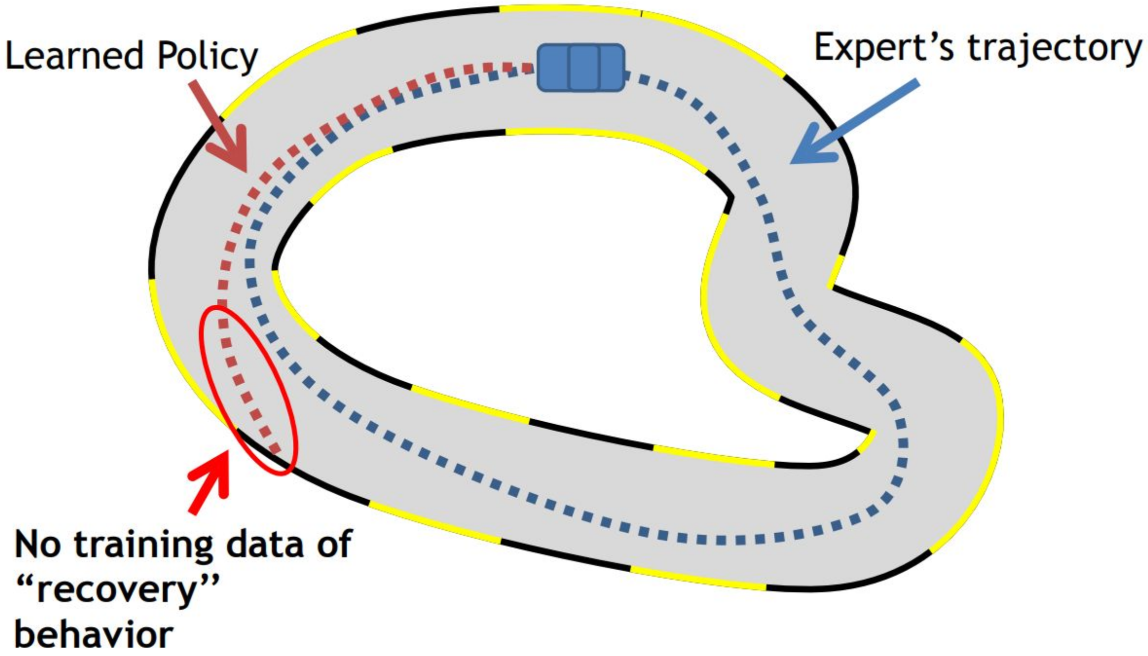
Behavior Cloning: Distribution Shift



Once we make a mistake, we end up in states which are not in the training data!

Credits: Wen Sun

Behavior Cloning: Distribution Shift

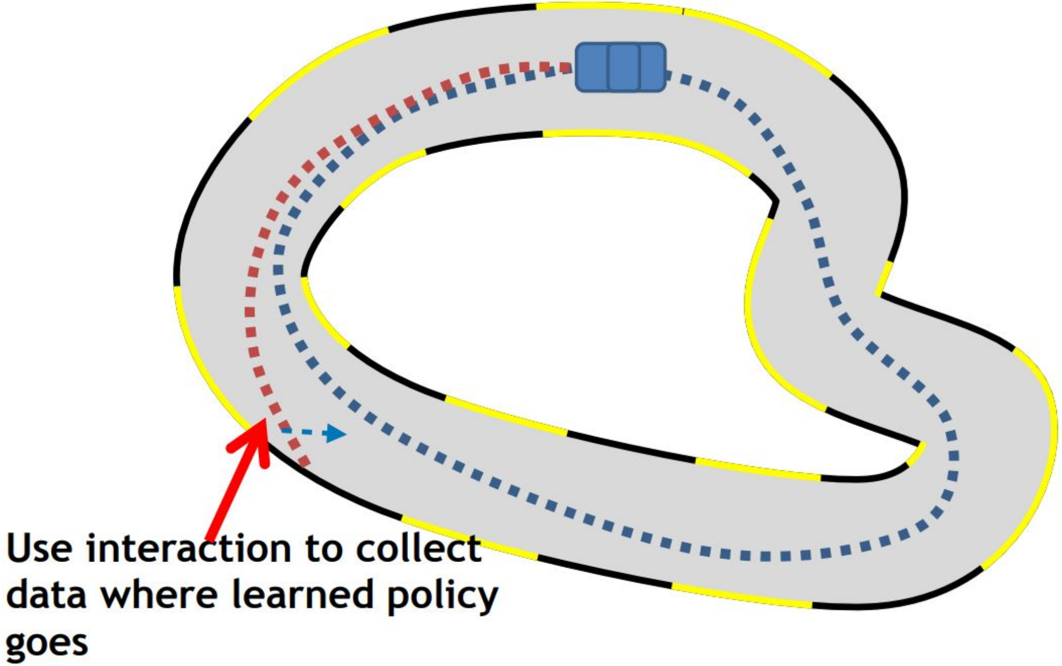


Credits: Wen Sun

Interactive Imitation Learning

Can we alleviate such problem? Yes, by setting up an interactive process where we continuously query the expert

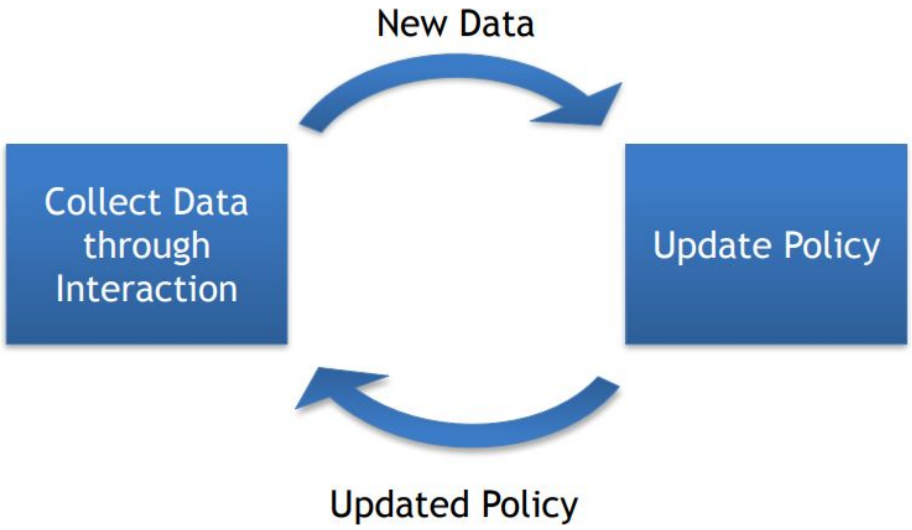
Interactive Imitation Learning



SAPIENZA UNIVERSITÀ DI ROMA

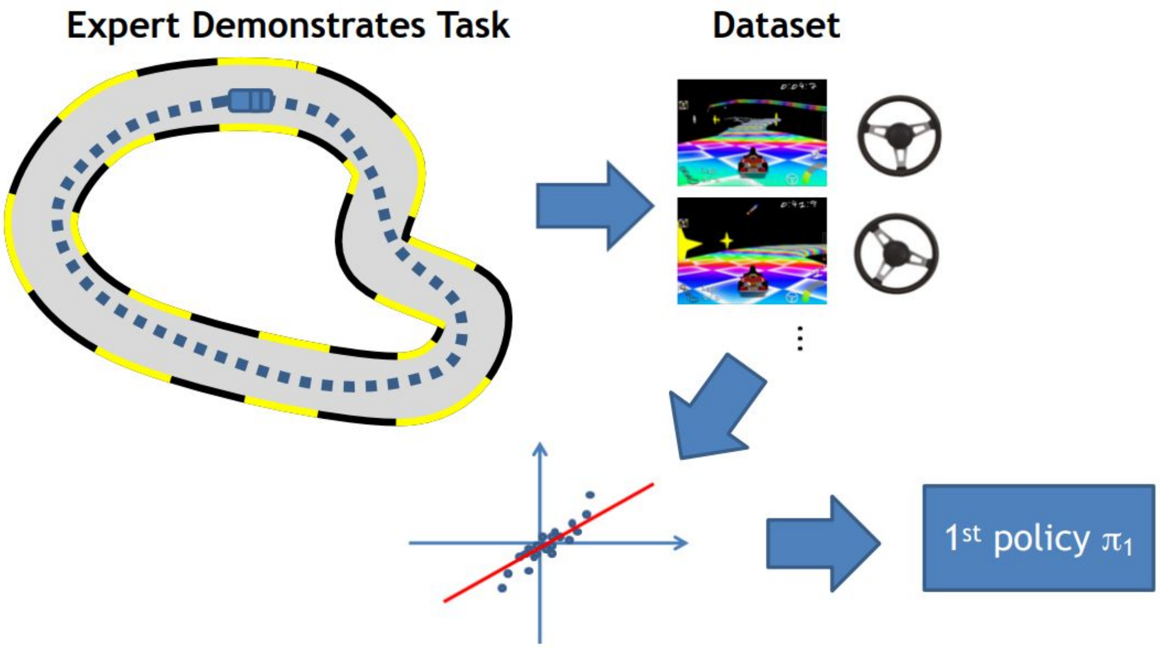
Credits: Wen Sun

DAgger



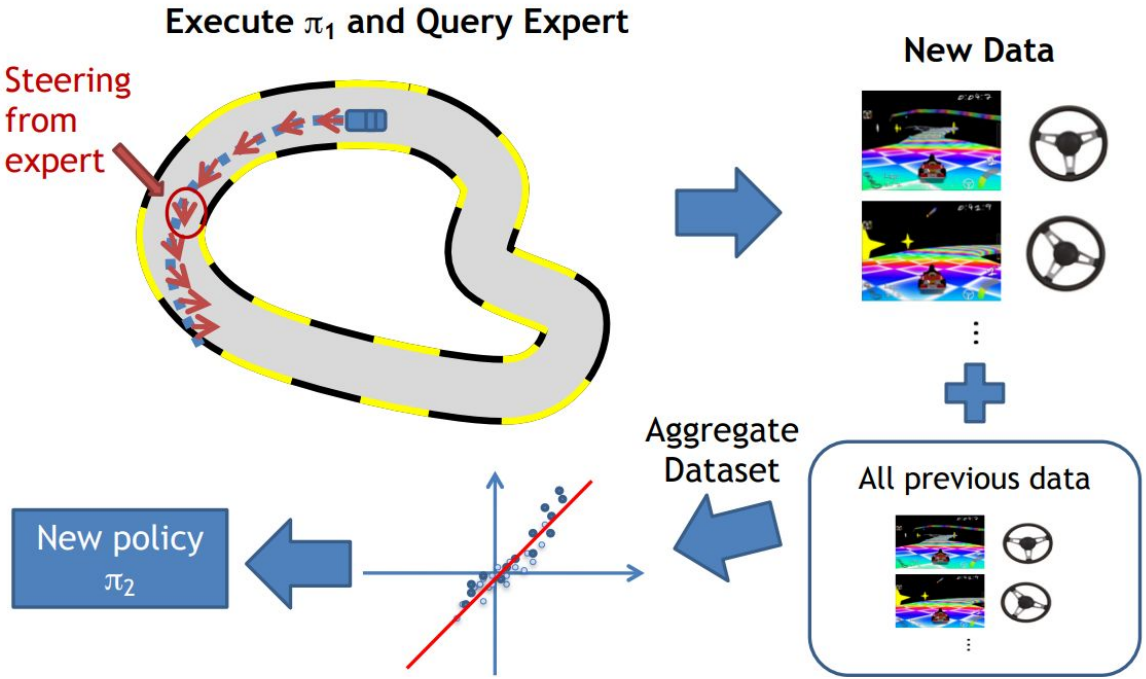
Credits: Wen Sun, Drew Bagnell, Stephane Ross, Arun Venktraman

DAgger: Iterations (0th)



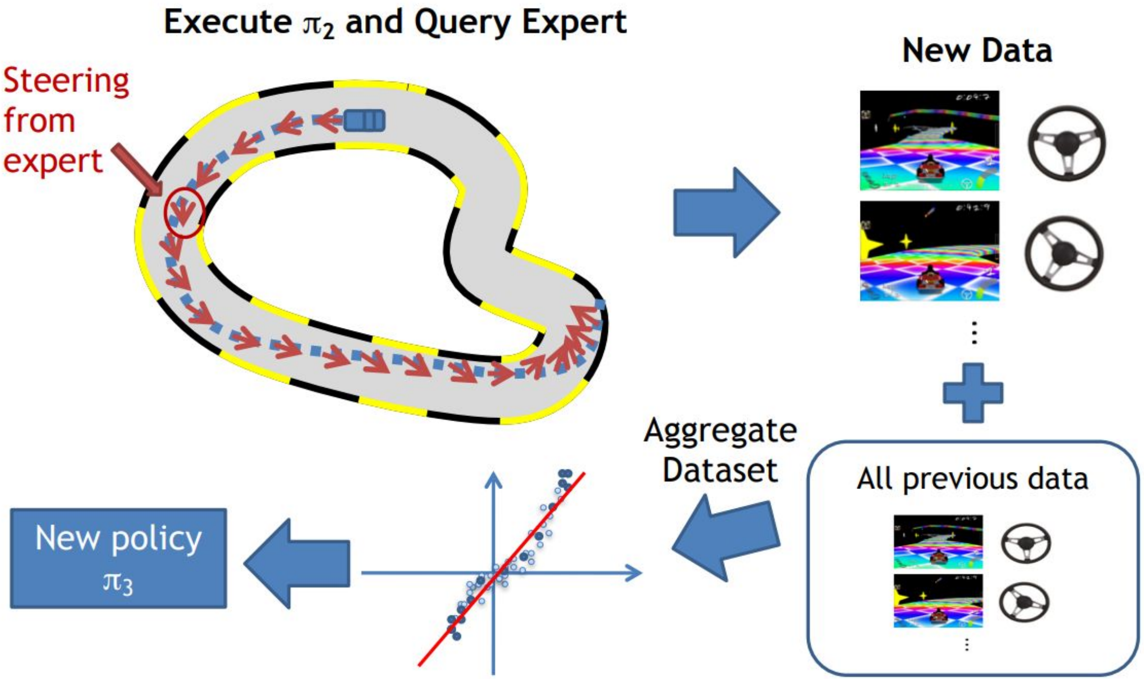
Credits: Wen Sun, Drew Bagnell, Stephane Ross, Arun Venktraman

DAgger: Iterations (1st)



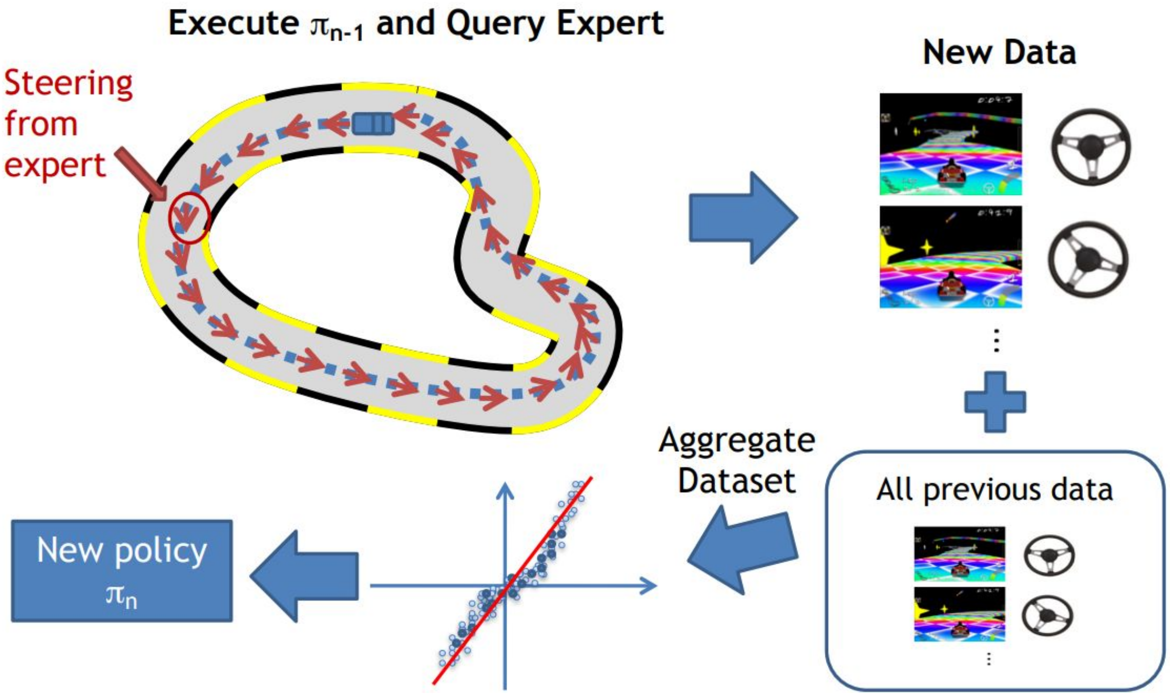
Credits: Wen Sun, Drew Bagnell, Stephane Ross, Arun Venktraman

DAgger: Iterations (2nd)



Credits: Wen Sun, Drew Bagnell, Stephane Ross, Arun Venktraman

DAgger: Iterations (n-th)



SAPIENZA UNIVERSITÀ DI ROMA

Credits: Wen Sun, Drew Bagnell, Stephane Ross, Arun Venktraman

DAgger: Video

Inverse Reinforcement Learning

Given an MDP, what happens if we cannot get access to a reward?

We can learn through an optimal expert that minimizes the true cost!

Assume transition function is known and we have our dataset *D*

Inverse Reinforcement Learning

Given an MDP, what happens if we cannot get access to a reward?

We can learn through an optimal expert that minimizes the true cost!

Assume transition function is known and we have our dataset *D*

Also assume the true reward/cost is linear in some features □(s,a)

Entropy

Given a distribution P, the entropy is:

$$\text{Entropy}(P) = -\sum_{x} P(x) \cdot \ln P(x)$$

Higher entropy means higher uncertainty (i.e., a deterministic distribution has 0 entropy, uniform the highest)

Maximum Entropy

We want to find a distribution whose mean and covariance matrix equal some values, but there are infinitely many such distributions:

we choose the least committing one, with maximum entropy

Maximum Entropy IRL

We want to find a policy such that

$$\max_{\pi} \mathbb{E}_{s\sim d_\mu^\pi}\left[\text{entropy}\left(\pi(\cdot\,|\,s)\right)\right]$$

$$s.t, \mathbb{E}_{s,a\sim d_\mu^\pi}\phi(s,a) = \boxed{\mathbb{E}_{s,a\sim d_\mu^{\pi^\star}}\phi(s,a)}$$

From expert data:

$$\sum_{i=1}^{N} \phi(s_i^\star, a_i^\star)/N$$

Maximum Entropy IRL

We want to find a policy such that

$$\max_{\pi} \mathbb{E}_{s\sim d_{\mu}^{\pi}} \boxed{\left[\text{entropy}\left(\pi(\cdot \mid s)\right)\right]}$$

$$s.t, \mathbb{E}_{s,a\sim d_{\mu}^{\pi}} \phi(s,a) = \mathbb{E}_{s,a\sim d_{\mu}^{\pi^{\star}}} \phi(s,a)$$

$$\mathbb{E}_{s\sim d_{\mu}^{\pi}} \left[\text{entropy}(\pi(\cdot \mid s))\right] = -\mathbb{E}_{s\sim d_{\mu}^{\pi}} \mathbb{E}_{a\sim\pi(\cdot\mid s)} \ln \pi(a \mid s) = -\mathbb{E}_{s,a\sim d_{\mu}^{\pi}} \ln \pi(a \mid s)$$

Maximum Entropy IRL

We want to find a policy such that

$$\boxed{\max_{\pi} \mathbb{E}_{s\sim d_\mu^\pi}\left[\text{entropy}\left(\pi(\cdot\,|\,s)\right)\right]}$$

$$s.t, \mathbb{E}_{s,a\sim d_\mu^\pi}\phi(s,a) = \mathbb{E}_{s,a\sim d_\mu^{\pi^\star}}\phi(s,a)$$

$$\arg\max_{\pi} \mathbb{E}_{s\sim d_\mu^\pi}\left[\text{entropy}(\pi(\cdot\,|\,s))\right] = \arg\min_{\pi} \mathbb{E}_{s,a\sim d_\mu^\pi} \ln \pi(a\,|\,s)$$

Maximum Entropy IRL

We want to find a policy such that

$$\boxed{\begin{aligned} &\max_{\pi} \mathbb{E}_{s \sim d_\mu^\pi} \left[\text{entropy} \left(\pi(\cdot \mid s) \right) \right] \\ &s.t, \mathbb{E}_{s,a \sim d_\mu^\pi} \phi(s,a) = \mathbb{E}_{s,a \sim d_\mu^{\pi^\star}} \phi(s,a) \end{aligned}}$$

Using Lagrange multipliers

$$\max_{w \in \mathbb{R}^d} \min_{\pi} \mathbb{E}_{s,a \sim d_\mu^\pi} \ln \pi(a \mid s) + w^\top \left(\mathbb{E}_{s,a \sim d_\mu^\pi} \phi(s,a) - \mathbb{E}_{s,a \sim d_\mu^{\pi^\star}} \phi(s,a) \right)$$

Maximum Entropy IRL: Algorithm

Initialize $w^0 \in \mathbb{R}^d$

This is like an RL problem w/ cost

For $t = 0 \to T - 1$

$c(s,a) := (w^t)^\top \phi(s,a)$, but w/ an additional $\ln \pi(a \mid s)$

$$\pi^t = \arg\min_{\pi} \mathbb{E}_{s,a \sim d_\mu^\pi} \left[(w^t)^\top \phi(x,a) + \ln \pi(a \mid s) \right]$$

(# best response: $\pi^t = \arg\min_{\pi} \ell(\pi, w^t)$)

$$w^{t+1} = w^t + \eta \left(\mathbb{E}_{s,a \sim d_\mu^{\pi^t}} \phi(s,a) - \mathbb{E}_{s,a \sim d_\mu^{\pi^\star}} \phi(s,a) \right)$$

(# gradient update: $w^{t+1} = w^t + \eta \nabla_w \ell(\pi^t, w^t)$)

Return $\bar{\pi} = \text{Uniform}(\pi^0, \ldots, \pi^{T-1})$

SAPIENZA UNIVERSITÀ DI ROMA

Credits: Wen Sun

Maximum Entropy IRL: Algorithm

Initialize $w^0 \in \mathbb{R}^d$

For $t = 0 \to T - 1$

This is like an RL problem w/ cost
$c(s,a) := (w^t)^\top \phi(s,a)$, but w/ an additional $\ln \pi(a \mid s)$

$$\pi^t = \arg\min_{\pi} \mathbb{E}_{s,a\sim d_\mu^\pi} \left[(w^t)^\top \phi(x,a) + \ln \pi(a \mid s) \right]$$

Uses soft-value iteration

(# best response: $\pi^t = \arg\min_{\pi} \ell(\pi, w^t)$)

$$w^{t+1} = w^t + \eta \left(\mathbb{E}_{s,a\sim d_\mu^{\pi^t}} \phi(s,a) - \mathbb{E}_{s,a\sim d_\mu^{\pi^\star}} \phi(s,a) \right)$$

(# gradient update: $w^{t+1} = w^t + \eta \nabla_w \ell(\pi^t, w^t)$)

Return $\bar{\pi} = \text{Uniform}(\pi^0, \ldots, \pi^{T-1})$