Introduction to inverse reinforcement learning (2/3)

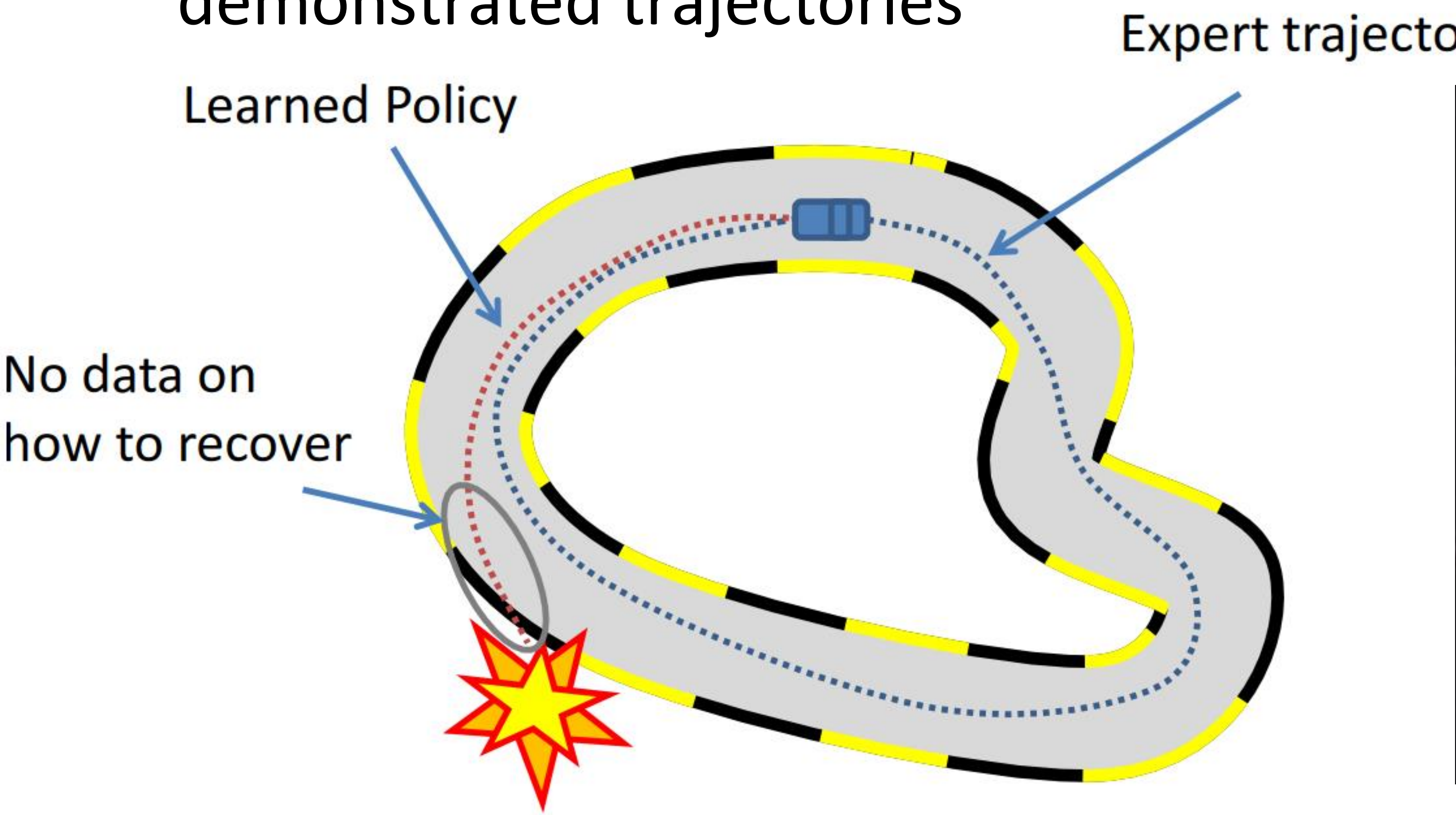
Eiji Uchibe

Dept. of Brain Robot Interface

ATR Computational Neuroscience Labs.

Naïve Imitation Learning Works Poorly

- The robot’s errors compound when drifting away from the expert’s demonstrated trajectories



Expert trajectory

Learned Policy

No data on how to recover

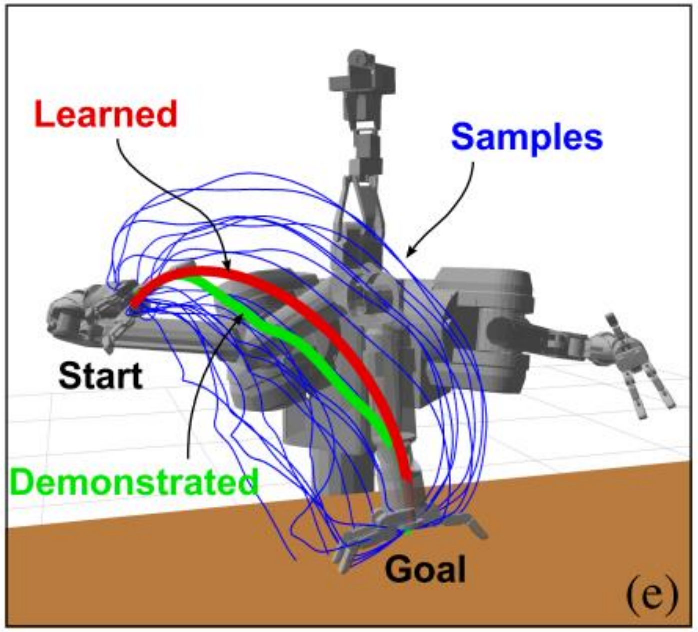
- No mechanism to recover when the generated trajectory deviates from the demonstrated ones

[Ross and Bagnell, 2010]

Preparation

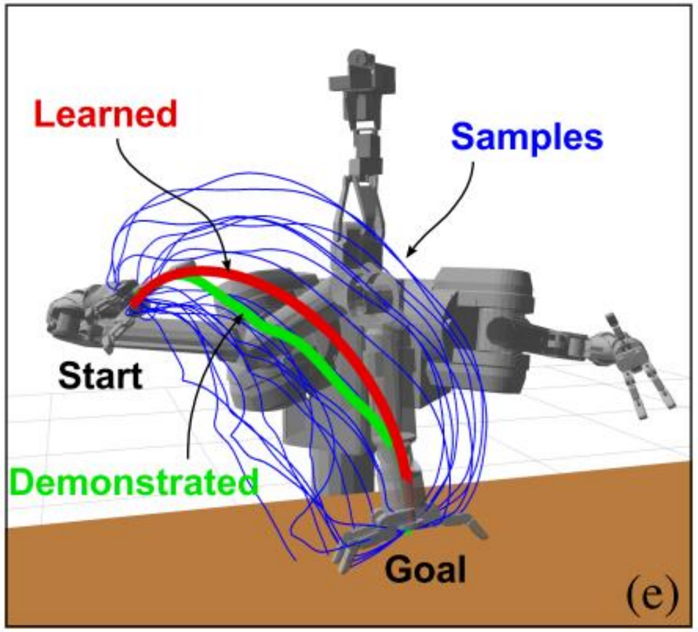
- $\tau = \{s_1, a_1, \ldots, s_T, a_T\}$: trajectory of state s and action a (green)
- $r(s, a; w)$: immediate reward at state s and action a parameterized by w
  - For a linear function approximator, $r(s, a; w) = w^\top \phi(s, a)$, where $\phi(s, a)$ is a feature vector
- Total reward along a trajectory τ

$$R(\tau) = \sum\nolimits_t r(s_t, a_t; w) = w^\top \sum\nolimits_t \phi(s_t, a_t) = w^\top \phi(\tau)$$



Kalakrishnan, M., Pastor, P., Righetti, L., & Schaal, S. (2013). Learning objective functions for manipulation. *Proc. of ICRA*, 1331–1336.

Maximum Entropy Inverse RL



- $P(\tau|w)$: probability of τ parameterized by w (blue)
- The expectation of features under $P(\tau|w)$ is equal to the expected empirical feature count

$$\underbrace{\sum_{\tau} P(\tau \mid w)\phi(\tau)}_{\text{expected feature under } P(\tau \mid w)} = \underbrace{\frac{1}{m}\sum_{j} \phi(\tau_j)}_{\text{expected empirical feature count of the expert}} \triangleq \tilde{\phi}$$

- Since many reward weights satisfy the above constraint, the principle of maximum entropy is employed to resolve ambiguities in choosing distributions

Ziebart, B.D., Maas, A., Bagnell, J.A., and Dey, A. (2008). Maximum entropy inverse reinforcement learning. *Proc. of AAAI*, 1433-1438.

Principle of maximum entropy (1/2)

- Choose a 'best' from a number of different probability distributions that all express the current state of knowledge
- Constrained optimization problem

$\max_{p} - \sum_{\tau} p(\tau) \ln p(\tau)$ → Maximize the entropy of $p(\tau)$

$\sum_{\tau} p(\tau)\phi_i(\tau) = \tilde{\phi}_i, \forall i$ → constraints on the expected feature

$\sum_{\tau} p(\tau) = 1,\ p(\tau) \geq 0$ → $p(\tau)$ is a probability distribution

Ziebart, B.D., Maas, A., Bagnell, J.A., and Dey, A. (2008). Maximum entropy inverse reinforcement learning. *Proc. of AAAI*, 1433-1438.

Principle of maximum entropy (2/2)

- The Lagrangian is given by

$$\mathcal{L} = -\sum\nolimits_{\tau} p(\tau) \ln p(\tau) + \lambda\left(\sum\nolimits_{\tau} p(\tau) - 1\right) + \sum\nolimits_{i} w_i \left(\sum\nolimits_{\tau} p(\tau)\phi_i(\tau) - \tilde{\phi}_i\right)$$

  - λ, w_i: Lagrange multipliers

- Calculus of variation: $\partial\mathcal{L}/\partial p = 0$ yields

$$\ln p(\tau) = \sum\nolimits_{i} w_i \phi_i(\tau) + \lambda - 1$$

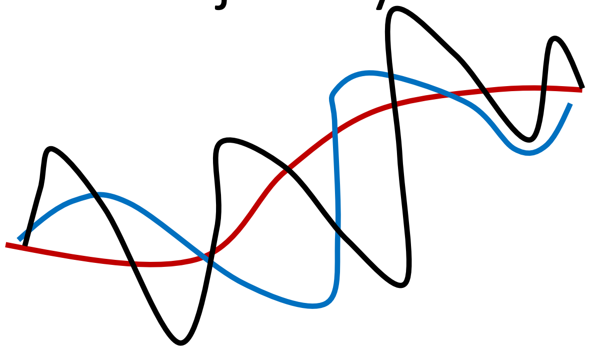
➡ $$p(\tau|w) = \frac{1}{Z(w)} \exp\left(w^{\top}\phi(\tau)\right) = \frac{1}{Z(w)} \exp\left(R(\tau)\right)$$

Probability under the principle of maximum entropy

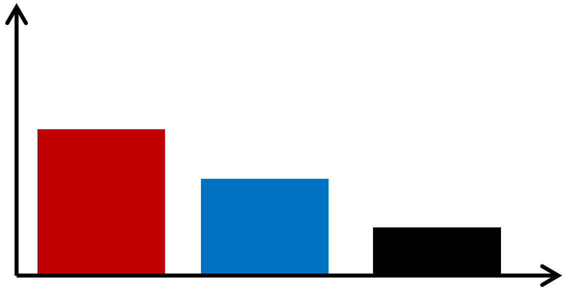
- Trajectories with equivalent returns have equal probabilities
- Trajectories with higher returns are exponentially more preferred

$$p(\tau|w) = \frac{1}{Z(w)} \exp\big(R(\tau; w)\big)$$

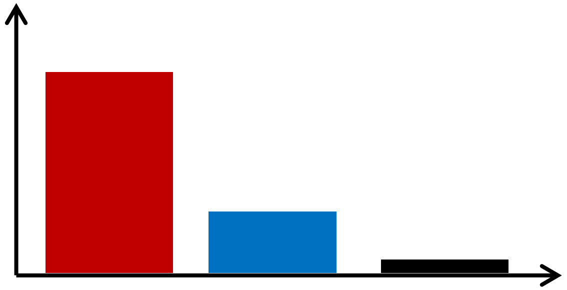
trajectory



$R(\tau_i)$ total rewards



$\Pr(\tau_i)$ probability



Ziebart, B.D., Maas, A., Bagnell, J.A., and Dey, A. (2008). Maximum entropy inverse reinforcement learning. *Proc. of AAAI*, 1433-1438.

Parameter optimization

- w is estimated by maximum likelihood from the expert's dataset

$$\mathcal{D}^{\pi} = \left\{\tau_j^{\pi}\right\}_{j=1}^{N}$$

- Log-likelihood is given by

$$L(w) = \frac{1}{N}\sum\nolimits_{\tau_j^{\pi}\in\mathcal{D}^{\pi}} \ln p\left(\tau_j^{\pi} \mid w\right) = \frac{1}{N}\sum\nolimits_{\tau_j^{\pi}\in\mathcal{D}^{\pi}} R(\tau_j^{\pi}; w) - \ln Z(w)$$

- w is updated by gradient ascent

$$w \leftarrow w + \alpha\nabla L(w), \qquad \nabla L(w) = \frac{1}{N}\sum\nolimits_{\tau_j^{\pi}\in\mathcal{D}^{\pi}} \nabla R\left(\tau_j^{\pi}; w\right) - \nabla \ln Z(w)$$

- How to evaluate the gradient of $\ln Z(\boldsymbol{w})$?

MaxEnt-IRL: evaluate $Z(w)$ by solving model-based reinforcement learning

- $$\nabla \ln Z(w) = \frac{1}{Z(w)} \frac{\partial Z(w)}{\partial w} = \frac{1}{Z(w)} \sum_{\tau} \exp\left(w^{\top}\phi(\tau)\right)\phi(\tau)$$
  $$= \underbrace{\sum_{\tau} P(\tau; w)\phi(\tau)}_{\text{expected feature under } P(\tau;w)} = \mathbb{E}_{P(\tau;w)}[\phi(\tau)]$$

- In RL, the following equation holds
  $$\sum_{\tau} P(\tau; w)\phi(\tau) = \underbrace{\sum_{s,a} P(s, a; w)\phi(s, a)}_{\text{expected feature under the joint state-action distribution}}$$

Samples from $P(s, a; w)$ can be generated by the optimal policy $\pi^*(a \mid s; w)$ trained with the current estimated reward $r(s, a; w)$

Simulation: FrozenLake task

- Discrete-state and discrete-action MDP task provided by OpenAI gym
  - Simplify the task by considering deterministic MDP
- Positive reward (+1) for entering the goal 🎁 from left
- 0 otherwise
- Episode ends when entering a hole 🔵
  - 0 reward is given to the agent
- https://uchibe.github.io/BILT-B/