Maximum Entropy Inverse Reinforcement Learning

Simon Bihlmaier

University of Stuttgart

st149710@stud.uni-stuttgart.de

July 14, 2021

Overview

Inverse Reinforcement Learning (IRL)

Given

- Markov Decision Process $(S, A, T, [r], \gamma)$,
- Dataset $\mathcal{D} = \{\tau_1, \tau_2, ...\}$
  of Expert Trajectories $\tau = ((s_1, a_1), (s_2, a_2), ...) \in \mathcal{D}$,

find the Reward Function $r(s)$ that explains the Expert Trajectories best.

This problem is not well-posed: There exist multiple Reward Functions that lead to the same optimal policy that generated the Expert Trajectories.

(Shannon) Entropy

Random variable X, probability distribution of outcomes $P(X = x)$

$$H(X) = -\sum_{x \in X} P(X = x) \log(P(X = x))$$

Intuition:
Highest value for uniform distribution $P(X = x) = \frac{1}{|X|}$:
This is the Maximum Entropy if there are no constraints on P

$$H(X) = -\sum_{x \in X} \frac{1}{|X|} \log(\frac{1}{|X|}) = -\log(1/|X|) = \log(|X|)$$

Lowest value for most "non-uniform" distribution $P(X = x_1) = 1$:

$$H(X) = 1 \cdot \log(1) = 0$$

Feature Expectation

Accumulated Trajectory Features:

$$f_\tau = \sum_{s_t \in \tau} f_{s_t}$$

Accumulated Trajectory Reward:

$$r_\theta(\tau) = \sum_{s_t \in \tau} r_\theta(s_t) = \sum_{s_t \in \tau} \theta^T f_{s_t} = \theta^T f_\tau$$

(Empirical) Accumulated Feature Expectation:

$$\tilde{f} = E_{P(\tau)}[f_\tau] = \sum_{\tau} P(\tau) f_\tau \approx \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} f_\tau$$

Optimization Problem

Maximize Entropy of the Trajectory Distribution resulting from reward function $r_\theta(s)$ while matching feature expectations to those of the expert demonstrations. [Ziebart, 2010]

$$\begin{aligned}
\arg\max_{\theta} \quad & H_{P(\tau|\theta,T)}(X) \\
\text{s.t.} \quad & \tilde{f}_P = \tilde{f}_E \\
& \sum_{\tau} P(\tau|\theta,T) = 1 \\
& P(\tau|\theta,T) \geq 0 \qquad \forall \tau \in X
\end{aligned}$$

Optimization Problem

The maximum entropy probability distribution satisfies the Maximization: [Ziebart, 2010]

$$P(\tau|\theta, T) = Z^{-1} e^{\theta^T f_\tau} = Z^{-1} e^{r_\theta(\tau)}$$

Where Z is called Partition Function:

$$Z = \sum_{\tau \in \mathcal{D}} e^{r_\theta(\tau)}$$

Optimization Problem

Solving previous optimization problem corresponds to maximizing the likelihood of the observed data under the maximum entropy distribution: [Ziebart, 2008]

$$\theta^* = \arg\max_{\theta} \sum_{\tau \in \mathcal{D}} \log P(\tau|\theta, T)$$

Gradient Update: Derivation from Log Likelihood

$$
\begin{aligned}
L &= \sum_{\tau \in \mathcal{D}} \log P(\tau | \theta, T) \\
&= \left(\sum_{\tau \in \mathcal{D}} \theta^T f_\tau\right) - |\mathcal{D}| \log \sum_{\tau \in \mathcal{D}} e^{\theta^T f_\tau} \\
\nabla_\theta L &= \left(\sum_{\tau \in \mathcal{D}} f_\tau\right) - |\mathcal{D}| \sum_{\tau \in \mathcal{D}} P(\tau | \theta, T) f_\tau \\
\frac{1}{|\mathcal{D}|} \nabla_\theta L &= \tilde{f} - \sum_{\tau} P(\tau | \theta, T) f_\tau \\
&= \tilde{f} - \sum_{s \in S} P(s | \theta, T) f_s \\
\theta_{t+1} &= \theta_t + \alpha \frac{1}{|\mathcal{D}|} \nabla_\theta L
\end{aligned}
$$

Solving for State Visitation Frequencies

Backward pass

1. Set $Z_{s_{\text{terminal}}} = 1$
2. Recursively compute for N iterations

$$Z_{a_{i,j}} = \sum_k P(s_k|s_i, a_{i,j}) e^{\text{reward}(s_i|\theta)} Z_{s_k}$$

$$Z_{s_i} = \sum_{a_{i,j}} Z_{a_{i,j}} + \mathbf{1}_{\{s_i = s_{\text{terminal}}\}}$$

Local action probability computation

3. $P(a_{i,j}|s_i) = \frac{Z_{a_{i,j}}}{Z_{s_i}}$

Forward pass

4. Set $D_{s_i,t} = P(s_i = s_{\text{initial}})$
5. Recursively compute for $t = 1$ to N

$$D_{s_k,t+1} = \sum_{s_i} \sum_{a_{i,j}} D_{s_i,t} P(a_{i,j}|s_i) P(s_k|a_{i,j}, s_i)$$

Summing frequencies

6. $D_{s_i} = \sum_t D_{s_i,t}$

[Ziebart, 2008]

Steps 1-3:
Computing stochastic Policy according to Maximum Entropy Distribution from current Reward Approximation
Similar to Bellman Equation:

$$v_\pi(s) = \sum_{a,s',r} P(s'|s,a)\pi(a|s)(\gamma v^\pi(s') + r)$$

Steps 4-6:
Computing State Visitation Frequencies from Policy

Algorithm

Initialize reward weights θ and collect Expert Trajectories $\mathcal{D}$

1. Calculate the Empirical Feature Expectation $\tilde{f}$ of the Expert Trajectories $\mathcal{D}$

Repeat:

2. Calculate a (stochastic) policy π according to the current approximation of the reward function
3. Calculate the State Visitation Frequencies $P(s|\theta, T)$ for π
4. Update the rewards weights θ by Gradient Ascend on the Log Likelihood of the Expert Trajectories:
   $\theta_{t+1} = \theta_t + \alpha \frac{1}{|\mathcal{D}|}(\tilde{f} - \sum_{s \in S} P(s|\theta, T) f_s)$

Finally:

5. Compute policy π from reward $r(s) = \theta^T f_s$ using Value Iteration

FrozenLake Environment

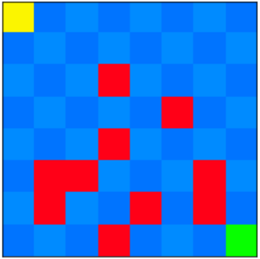

Figure: Frozen Lake 8x8 environment

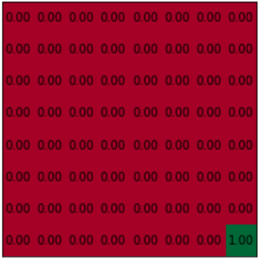

Figure: True Reward Function

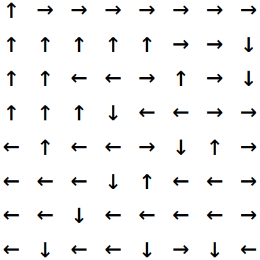

Figure: Expert Policy from True Reward

Reward Function Comparison

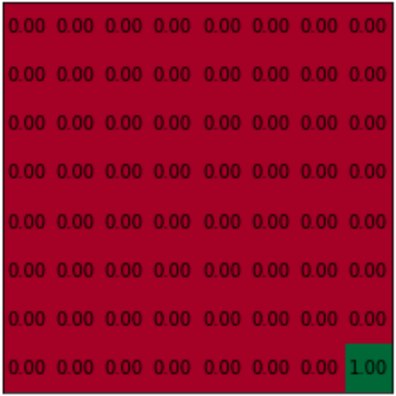

Figure: True Reward Function

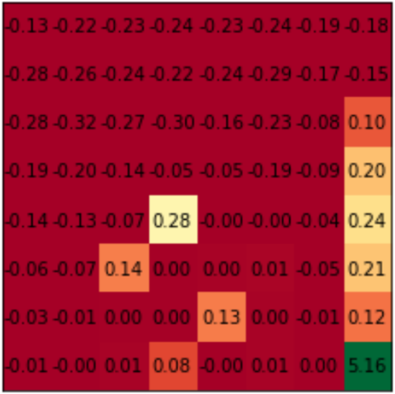

Figure: Learned Reward Function

Policy Comparison

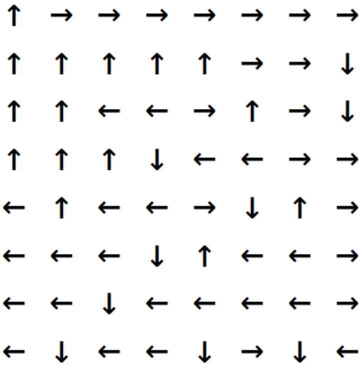

Figure: Expert Policy: 0.85 Successes per Episode

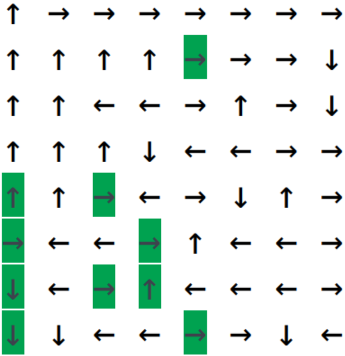

Figure: Policy derived from learned Reward: 0.81 Successes per Episode

Video

Limitations of Maximum Entropy IRL

- Only suitable for discrete State and Action spaces
- System Dynamics $P(s'|s, a)$ are required for forward and backward pass
- Linear Reward Function limits expressiveness: Feature choice is important
- Non-deterministic Transitions are only allowed to have limited impact on agent behaviour

References

Brian D. Ziebart (2008)
Maximum Entropy Inverse Reinforcement Learning
Authors: Brian D. Ziebart and Andrew Maas and J. Andrew Bagnell and Anind K. Dey
Proc. AAAI, pages 1433-1438

Brian D. Ziebart (2010)
Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy
Dissertation, Carnegie Mellon University Pittsburgh, PA 15213

Questions

Gradient Update: Derivation from Log Likelihood

$$
\begin{aligned}
L &= \sum_{\tau \in \mathcal{D}} \log P(\tau | \theta, T) \\
&= \sum_{\tau \in \mathcal{D}} \log \frac{e^{\theta^T f_\tau}}{Z} \\
&= \sum_{\tau \in \mathcal{D}} (\theta^T f_\tau - \log Z) \\
&= (\sum_{\tau \in \mathcal{D}} \theta^T f_\tau) - |\mathcal{D}| \log Z \\
&= (\sum_{\tau \in \mathcal{D}} \theta^T f_\tau) - |\mathcal{D}| \log \sum_{\tau \in \mathcal{D}} e^{\theta^T f_\tau} \\
\nabla_\theta L &= (\sum_{\tau \in \mathcal{D}} f_\tau) - |\mathcal{D}| (\sum_{\tau \in \mathcal{D}} e^{\theta^T f_\tau})^{-1} \sum_{\tau \in \mathcal{D}} f_\tau e^{\theta^T f_\tau} \\
&= (\sum_{\tau \in \mathcal{D}} f_\tau) - |\mathcal{D}| \sum_{\tau \in \mathcal{D}} P(\tau | \theta, T) f_\tau
\end{aligned}
$$

Gradient Update: State Visitation Frequencies

$$\nabla_\theta L = (\sum_{\tau \in \mathcal{D}} f_\tau) - |\mathcal{D}| \sum_\tau P(\tau|\theta, T) f_\tau$$

$$\frac{1}{|\mathcal{D}|} \nabla_\theta L = \frac{1}{|\mathcal{D}|} (\sum_{\tau \in \mathcal{D}} f_\tau) - \sum_\tau P(\tau|\theta, T) f_\tau$$

$$= \tilde{f} - \sum_\tau P(\tau|\theta, T) f_\tau$$

$$= \tilde{f} - \sum_{s \in S} P(s|\theta, T) f_s$$

$$\theta_{t+1} = \theta_t + \alpha \frac{1}{|\mathcal{D}|} \nabla_\theta L$$