Connections Between Inference And Control

Aaron Lou

July 2018

1 Optimal Control and Human Behavior

Optimal control is very similar to Human Behavior (try to optimize reward based on model dynamics). In particular

$$a_1, \dots, a_T = \arg \max_{a_1, \dots a_T} \sum_{t=1}^{T} r(s_t, a_t) \quad s_{t+1} = f(s_t, a_t)$$

$$\pi = \arg \max_{\pi} E_{s_{t+1} \sim p(s_{t+1} | s_t, a_t), a_t \sim \pi(a_t | s_t)}[r(s_t, a_t)] \quad a_t \sim \pi(a_t \mid s_t)$$

However, behavior is stochastic and there always get errors. For example, they give the example of a monkey trying to manuever a dot to get juice. Some mistakes (being correct) matter more than being optimal to the degree of a couple of milliseconds, but good behavior is still the most likely.

1.1 A Probabilistic Graphical Model of Decision Making

If we set $\tau = p(s_{1:T}, a_{1:T})$ to be the probability of seeing trajectories (just physics and likely actions), this makes no assumption on optimal behavior. If we add $\mathcal{O}$ to the MDP which represents "the monkey trying to optimally achieve a goal" and set this to be binary (0- not trying, 1-trying).

We now attempt to solve the inference problem $p(\tau \mid \infty : \mathcal{T})$ or when the monkey is trying to be optimal and need to pick (or assume that) $p(\mathcal{O} \mid s_t, a_t) \propto \exp(r(s_t, a_t))$ which means that you're more likely to be trying to achieve optimality if there is an immediate reward. If we do that

$$p(\tau \mid \mathcal{O}_{1:T}) = \frac{p(\tau, \mathcal{O}_{1:T})}{p(\mathcal{O}_{1:T})} \propto p(\tau) \prod_t \exp(r(s_t, a_t)) = p(\tau) \exp\left(\sum_t r(s_t, a_t) \right)$$

or the most rewarding trajectory which is feasible is the most likely.

1.2 Applications

This is interesting because it can

- Can model suboptimal behavior (inverse RL)
- Can apply inference models to solve control and planning models.
- Provides an explanation for why stochastic behavior might be preferred (exploration and transfer learning).

To do inference (which is planning), we note that we must have

1. Compute Backward Messages (calculate the probability of being optimal in the future) $\beta_t(s_t, a_t) = p(\mathcal{O}_{t:T} \mid s_t, a_t)$.
2. Compute policy $p(a_t \mid s_t, \mathcal{O}_{t:T}$ or the pathway when we assume optimality.
3. Calculate Forward messages (the probability of being in a state given optimality) $\alpha_t(s_t) = p(s_t \mid \mathcal{O}_{t:T})$.

1.2.1 Backward Message

The backward message β_t message gives the probability $\mathcal{O}$ is 1 from the present until the future. We can compute them from the back to the front of the trajectory (working forwards). Note that

$$p(\mathcal{O}_T \mid s_t, a_t) \propto \exp(r(s_t, a_t))$$

$$\beta_t(s_t, a_t) = p(\mathcal{O}_{t:T} \mid s_t, a_t) = \int p(\mathcal{O}_{t:T}, s_{t+1} \mid s_t, a_t) ds_{t+1} \quad t = [T-1]$$

$$= \int p(\mathcal{O}_{t+1:T} \mid s_{t+1}) p(s_{t+1} \mid s_t, a_t) p(\mathcal{O}_t \mid s_t, a_t) ds_{t+1}$$

the last two terms are (respectively) the dynamics and the reward model. However, for the first term, we have

$$p(\mathcal{O}_{t+1:T} \mid s_{t+1}) \int p(\mathcal{O}_{t+1} \mid s_{t+1}, a_{t+1}) p(a_{t+1} \mid s_{t+1}) da_{t+1}$$

and the first term is just $\beta_{t+1}(s_{t+1}, a_{t+1})$ and the second term is not dependent on optimality (so it can just be a uniform distribution and can be ignored). The algorithm is

$$\beta(s_t, a_t) = p(\mathcal{O} \mid s_t, a_t) E_{st+1 \sim p(s_{t+1} \mid s_t, a_t)}[\beta_{t+1}(s_{t+1})]$$

$$\beta_t(s_t) = E_{a_t \sim p(a_t \mid s_t)}[\beta_t(s_t, a_t)]$$

If let $V_t(s_t) = \log \beta_t(s_t)$ and $Q_t(s_t, a_t) = \log \beta_t(s_t, a_t)$. Then we find that

$$V_t(s_t) = \log \int \exp(Q_t(s_t, a_t)) da_t \quad V_t(s_t) \to \max_{a_t} Q_t(s_t, a_t) \text{ as } Q_t(s_t, a_t) \to \infty$$

We note that $Q_t(s_t, a_t)$ is similar to our Q-function and $Q_t(s_t, a_t) = r(s_t, a_t) + \log E[\exp(V_{t+1}(s_{t+1}))]$ so we have $Q_t(s_t, a_t) = r(s_t, a_t) + V_{t+1}(s_{t+1})$ and the stochastic case is $Q_t(s_t, a_t) = r(s_t, a_t) + E[V_{t+1}(s_{t+1})]$ and is the "optimistic outlook" on dynamics. This isn't always correct (there is a correction paper).

The action prior $p(a_{t+1} \mid s_{t+1})$ was assumed to be uniform. The value equation becomes

$$V(s_t) = \log \int \exp(Q(s_t, a_t) + \log p(a_t \mid s_t)) da_t$$

$$Q(s_t, a_t) = r(s_t, a_t) + E[V(s_{t+1})]$$

Letting $\tilde{Q}(s_t, a_t) = r(s_t, a_t) + \log p(a_t \mid s_t) + E[V(s_{t+1})]$ and we are just taking a "soft max" (maximum as $Q \to \infty$) over $\tilde{Q}$ so we can always disregard the action prior term WLOG.

1.2.2 Computing Policy

Now we attempt to compute the policy $p(a_t \mid s_t, \mathcal{O}_{t:T}) = \pi(a_t \mid s_t)$. We find that

$$p(a_t \mid s_t, \mathcal{O}_{t:T}) = \frac{p(a_t, s_t \mid \mathcal{O}_{t:T})}{p(s_t \mid \mathcal{O}_{t:T})} = \frac{p(\mathcal{O}_{t:T} \mid a_t, s_t) p(a_t, s_t) / p(\mathcal{O}_{t:T})}{p(\mathcal{O}_{t:T} \mid s_t) p(s_t) / p(\mathcal{O}_{t:T})}$$

$$= \frac{p(\mathcal{O}_{t:T} \mid a_t, s_t)}{p(\mathcal{O}_{t:T} \mid s_t)} \frac{p(a_t, s_t)}{p(s_t)} = \frac{\beta_t(s_t, a_t)}{\beta_t(s_t, a_t)} \beta_t(s_t) = \pi(a_t \mid s_t)$$

Recall our values of V, Q

$$V_t(s_t) = \log \beta_t(s_t) \quad Q_t(s_t, a_t) = \log \beta_t(s_t, a_t)$$

we have

$$\pi(s_t \mid a_t) = \exp(Q_t(s_t, a_t) - V_t(s_t)) = \exp(A_t(s_t, a_t))$$

Some variants include the discounted SOC: $Q_t(s_t, a_t) = r(s_t, a_t) + \gamma E[V_{t+1}(s_{t+1})]$ and explicit temperature: $V_t(s_t) = \alpha \log \int \exp(\frac{1}{\alpha} Q_t(s_t, a_t)) da_t$, or $\pi(s_t \mid a_t) = \exp(\frac{1}{\alpha} A_t(s_t, a_t))$.

This makes sense, as better actions are more probable with random tiebreaking. This is also analogous to Boltzmann exploration (and approaches greedy policy as temperature decreases).

1.2.3 Forward Messages

Recall that the forward message is $\alpha_t(s_t) = p(s_t \mid \mathcal{O}_{1:t-1})$. note that we have

$$\alpha_t(s_t) = \int p(s_t \mid s_{t-1}, a_{t-1}) p(a_{t-1} \mid s_{t-1}, \mathcal{O}_{t-1}) p(s_{t-1} \mid \mathcal{O}_{1:t-2}) ds_{t-1} da_{t-1}$$

and we note the first term is just the dynamics, the last term is the $\alpha_{t-1}(s_{t-1}$. The middle state can be simplified

$$p(a_{t-1} \mid s_{t-1}, \mathcal{O}_{t-1}) = \frac{p(\mathcal{O}_{t-1} \mid s_{t-1}, a_{t-1}) p(a_{t-1} \mid s_{t-1})}{p(\mathcal{O}_{t-1} \mid s_{t-1})}$$

Where is the first top term exponential reward, the second is the action prior, and the bottom term is a normalizer. If we want $p(s_t \mid \mathcal{O}_{t:T}$, we note that

$$p(s_t \mid \mathcal{O}_{t:T}) = \frac{p(s_t, \mathcal{O}_{t:T})}{p(\mathcal{O}_{t:T})} = \frac{p(\mathcal{O}_{t:T} \mid s_t) p(s_t, \mathcal{O}_{1:t-1})}{p(\mathcal{O}_{1:T})}$$
$$\propto \beta_t(s_t) p(s_t \mid \mathcal{O}_{1:t-1} p(\mathcal{O}_{1:t-1} \propto \beta_t(s_t) \alpha_t(s_t)$$

They are connected as follows. Imagining a cone projected from the last state to the first states representing all states that have a high probability of reaching the goal. Similarly, a cone from the start state projects the states that are easy to reach. The state marginals are the regions in the intersection between these two.

2 Reinforcement Learning with soft optimality

2.1 Q - Learning with Soft Optimality

Standard Q learning is given by

$$\phi \leftarrow \phi + \alpha \nabla_\phi Q_\phi(s, a)(r(s, a) + \gamma V(s') - Q_\phi(s, a))$$

Where the target is given by $V(s') = \max_{a'} Q_\phi(s', a')$. The soft Q learning is given by the same equation

$$\phi \leftarrow \phi + \alpha \nabla_\phi Q_\phi(s, a)(r(s, a) + \gamma V(s') - Q_\phi(s, a))$$

where $V(s') = \text{soft}\max_{a'} Q_\phi(s', a') = \log \int \exp(Q_\phi(s', a'))da'$. And the policy is given by $\pi(a \mid s) = \exp(Q_\phi(s, a) - V(s)) = \exp(A(s, a))$. The algorithm is given by

Result: Optimal Policy
while *until convergence* **do**
1. take some action a_i and observe (s_i, a_i, s'_i, r_i) and add to replay buffer $\mathcal{R}$.
2. sample mini-batch $\{s_j, a_j, s'_j, r_j\}$ from $\mathcal{R}$ uniformly.
3. compute $y_j = r_j + \gamma \text{soft}\max_{a'_j} Q_{\phi'}(s'_j, a'_j)$ using target network Q_ϕ.
4. $\phi \leftarrow \phi - \alpha \sum_j \frac{dQ_\phi}{d\phi}(s_j, a_j)(Q_\phi(s_j, a_j) - y_j)$.
5. update ϕ': copy ϕ every N steps, or Polyak average $\phi' \leftarrow \tau\phi' + (1-\tau)\phi$.

end

Algorithm 1: Soft Q Learning

2.2 Policy Gradient with Soft Optimality

Note that $\pi(a \mid s) = \exp(Q_\phi(s,a) - V(s))$ optimizes $\sum_t E_{\pi(s_t,a_t)}[r(s_t,a_t)] + E_{\pi(s_t)}[\mathcal{H}(\pi(a_t \mid s_t))]$. Where the second term is the entropy of the policy.

The intuition behind this is that $\pi(a \mid s) \propto \exp(Q_\phi(s,a))$ when π minimizes $D_{KL}(\pi(a \mid s) \mid\mid \frac{1}{Z}\exp(Q(s,a)))$ where

$$D_{KL}(\pi(a \mid s) \mid\mid \frac{1}{Z}\exp(Q(s,a))) = E_{\pi(a|s)}[Q(s,a)] - \mathcal{H}(\pi)$$

and this often called "entropy regularized" policy gradient and combats premature collapse (so the policy gradient can learn).

2.3 Policy Gradient vs Q-Learning

Policy gradient derivation seeks to maximize:

$$J(\theta) = \sum_t E_{\pi(s_t,a_t)}[r(s_t,a_t)] + E_{\pi(s_t)}[\mathcal{H}(\pi(a \mid s_t))] = \sum_t E_{\pi(s_t,a_t)}[r(s_t,a_t) - \log\pi(a_t \mid s_t)]$$

and our gradient is

$$\nabla_\theta \left[\sum_t E_{\pi(s_t,a_t)}[r(s_t,a_t) - \log\pi(a_t \mid s_t)]\right]$$

$$\approx \frac{1}{N}\sum_i \sum_t \nabla_\theta \log\pi(a_t \mid s_t)\left(r(s_t,a_t) + \left(\sum_{t'=t+1}^{T} r(s_{t'},a_{t'}) - \log\pi(a_{t'} \mid s_{t'})\right) - \log\pi(a_t \mid s_t) - 1\right)$$

If we recall that $\log\pi(s_t \mid a_t) = Q(s_t,a_t) - V(s_t)$ since we take that as the baseline, the function is

$$\approx \frac{1}{N}\sum_i \sum_t (\nabla_\theta Q(a_t \mid s_t) - \nabla_\theta V(s_t))(r(s_t,a_t) + Q(s_{t+1},a_{t+1}) - Q(s_t,a_t) + V(s_t))$$

and we can ignore the $V(s_t)$ since it is the baseline. The soft Q-learning gradient is

$$-\frac{1}{N}\sum_i\sum_t \nabla_\theta Q(a_t \mid s_t)\Big(r(s_t,a_t)+\operatorname*{softmax}_{a_{t+1}} Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\Big)$$

3 Benefits of Soft Optimality

- Improve exploration and prevent entropy collapse
- Easier to specialize (finetune) policies for more specific tasks
- Principled approach to break ties
- Better robustness (due to wider coverage of states)
- Can reduce hard optimality as temperature goes to 0.
- Can model human behavior (inverse RL)