Connections Between Inference and Control

CS 294-112: Deep Reinforcement Learning
Sergey Levine
Class Notes

1. Homework 3 is due today, at 11:59 pm
2. Homework 4 comes out tonight
3. Final project proposal due on Monday!
Today’s Lecture

1. Does reinforcement learning and optimal control provide a reasonable model of human behavior?
2. Is there a better explanation?
3. Can we derive optimal control, reinforcement learning, and planning as \textit{probabilistic inference}?
4. How does this change our RL algorithms?
5. (next week) We’ll see this is crucial for \textit{inverse} reinforcement learning

\textbullet Goals:
\begin{itemize}
\item Understand the connection between inference and control
\item Understand how specific RL algorithms can be instantiated in this framework
\item Understand why this might be a good idea
\end{itemize}
Optimal Control as a Model of Human Behavior

\[\mathbf{a}_1, \ldots, \mathbf{a}_T = \arg \max_{\mathbf{a}_1, \ldots, \mathbf{a}_T} \sum_{t=1}^{T} r(s_t, a_t)\]

\[s_{t+1} = f(s_t, a_t)\]

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

\[a_t \sim \pi(a_t|s_t)\]
What if the data is not optimal?

some mistakes matter more than others!

behavior is stochastic

but good behavior is still the most likely
A probabilistic graphical model of decision making

\[ a_1, \ldots, a_T = \arg \max_{a_1, \ldots, a_T} \sum_{t=1}^{T} r(s_t, a_t) \]

\[ s_{t+1} = f(s_t, a_t) \]

\[ p(s_{1:T}, a_{1:T}) = ?? \quad \text{no assumption of optimal behavior!} \]

\[ p(\tau|\mathcal{O}_{1:T}) \]

\[ p(\mathcal{O}_{t}|s_t, a_t) \propto \exp(r(s_t, a_t)) \]

\[ p(\tau|\mathcal{O}_{1:T}) = \frac{p(\tau, \mathcal{O}_{1:T})}{p(\mathcal{O}_{1:T})} \]

\[ \propto p(\tau) \prod_t \exp(r(s_t, a_t)) = p(\tau) \exp \left( \sum_t r(s_t, a_t) \right) \]
Why is this interesting?

- Can model suboptimal behavior (important for inverse RL)
- Can apply inference algorithms to solve control and planning problems
- Provides an explanation for why stochastic behavior might be preferred (useful for exploration and transfer learning)
Inference = planning

\[ p(O_t|s_t, a_t) \propto \exp(r(s_t, a_t)) \]
\[ p(s_{t+1}|s_t, a_t) \]

how to do inference?

1. compute backward messages \( \beta_t(s_t, a_t) = p(O_{t:T}|s_t, a_t) \)
2. compute policy \( p(a_t|s_t, O_{1:T}) \)
3. compute forward messages \( \alpha_t(s_t) = p(s_t|O_{1:t-1}) \)
Backward messages

\[ p(O_t | s_t, a_t) \propto \exp(r(s_t, a_t)) \]

\[ p(s_{t+1} | s_t, a_t) \]

\[ \beta_t(s_t, a_t) = p(O_{t:T} | s_t, a_t) \]

\[ = \int p(O_{t:T}, s_{t+1} | s_t, a_t) ds_{t+1} \]

\[ = \int p(O_{t+1:T} | s_{t+1}) p(s_{t+1} | s_t, a_t) p(O_t | s_t, a_t) ds_{t+1} \]

\[ \text{for } t = T - 1 \text{ to } 1: \]

\[ \beta_t(s_t, a_t) = p(O_t | s_t, a_t) E_{s_{t+1} \sim p(s_{t+1} | s_t, a_t)} [\beta_{t+1}(s_{t+1})] \]

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

which actions are likely *a priori*

(assume uniform for now)
A closer look at the backward pass

\[
\begin{align*}
\text{for } t &= T - 1 \text{ to } 1: \\
\beta_t(s_t, a_t) &= p(o_t | s_t, a_t) E_{s_{t+1} \sim p(s_{t+1} | s_t, a_t)}[\beta_{t+1}(s_{t+1})] \\
\beta_t(s_t) &= E_{a_t \sim p(a_t | s_t)}[\beta_t(s_t, a_t)] \\
\text{let } V_t(s_t) &= \log \beta_t(s_t) \\
\text{let } Q_t(s_t, a_t) &= \log \beta_t(s_t, a_t) \\
V_t(s_t) &= \log \int \exp(Q_t(s_t, a_t))da_t \\
V_t(s_t) &\rightarrow \max_{a_t} Q_t(s_t, a_t) \text{ as } Q_t(s_t, a_t) \text{ gets bigger!}
\end{align*}
\]

value iteration algorithm:

1. set \( Q(s, a) \leftarrow r(s, a) + \gamma E[V(s')] \)
2. set \( V(s) \leftarrow \max_a Q(s, a) \)

“optimistic” transition
(not a good idea!)

\[
Q_t(s_t, a_t) = r(s_t, a_t) + \log E[\exp(V_{t+1}(s_{t+1}))]
\]

deterministic transition: \( Q_t(s_t, a_t) = r(s_t, a_t) + V_{t+1}(s_{t+1}) \)

a better stochastic model: \( Q_t(s_t, a_t) = r(s_t, a_t) + E[V_{t+1}(s_{t+1})] \)

Ziebart et al. ‘10 “Modeling Interaction via the Principle of Maximum Causal Entropy”
Backward pass summary

\[ \beta_t(s_t, a_t) = p(O_{t:T} | s_t, a_t) \]

probability that we can be optimal at steps \( t \) through \( T \) given that we take action \( a_t \) in state \( s_t \)

for \( t = T - 1 \) to 1:

\[ \beta_t(s_t, a_t) = p(O_t | s_t, a_t) E_{s_{t+1} \sim p(s_{t+1} | s_t, a_t)} [\beta_{t+1}(s_{t+1})] \]

compute recursively from \( t = T \) to \( t = 1 \)

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

let \( V_t(s_t) = \log \beta_t(s_t) \)

let \( Q_t(s_t, a_t) = \log \beta_t(s_t, a_t) \)

log of \( \beta_t \) is “Q-function-like”
The action prior

remember this?

$$p(O_{t+1:T}|s_{t+1}) = \int p(O_{t+1:T}|s_{t+1}, a_{t+1}) p(a_{t+1}|s_{t+1}) da_{t+1}$$

$$\beta_t(s_{t+1}, a_{t+1})$$

(“soft max”)

what if the action prior is not uniform?

$$V(s_t) = \log \int \exp(\tilde{Q}(s_t, a_t) + \log p(a_t|s_t)) a_t$$

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

let $$\tilde{Q}(s_t, a_t) = r(s_t, a_t) + \log p(a_t|s_t) + E[V(s_{t+1})]$$

$$V(s_t) = \log \int \exp(\tilde{Q}(s_t, a_t)) a_t \Leftrightarrow V(s_t) = \log \int \exp(Q(s_t, a_t) + \log p(a_t|s_t)) a_t$$

can always fold the action prior into the reward! uniform action prior can be assumed without loss of generality
Policy computation

\[ p(O_t|s_t, a_t) \propto \exp(r(s_t, a_t)) \]
\[ p(s_{t+1}|s_t, a_t) \]

2. compute policy  \( p(a_t|s_t, O_{1:T}) \)

\[
p(a_t|s_t, O_{1:T}) = \pi(a_t|s_t)
\]
\[
= \frac{p(a_t, s_t|O_{t:T})}{p(s_t|O_{t:T})}
\]
\[
= \frac{p(O_{t:T}|a_t, s_t)p(a_t, s_t)/p(O_{t:T})}{p(O_{t:T}|s_t)p(s_t)/p(O_{t:T})}
\]
\[
= \frac{p(O_{t:T}|a_t, s_t)p(a_t, s_t)}{p(O_{t:T}|s_t)p(s_t)} = \frac{\beta_t(s_t, a_t) \pi(a_t|s_t)}{\beta_t(s_t)}
\]

\[
\beta_t(s_t, a_t) = p(O_{t:T}|s_t, a_t)
\]
\[
\beta_t(s_t) = p(O_{t:T}|s_t)
\]
\[
\pi(a_t|s_t) = \frac{\beta_t(s_t, a_t)}{\beta_t(s_t)}
\]
Policy computation with value functions

for $t = T - 1$ to 1:

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

$$V_t(s_t) = \log \int \exp(Q_t(s_t, a_t)) a_t$$

$$\pi(a_t | s_t) = \frac{\beta_t(s_t, a_t)}{\beta_t(s_t)} \quad V_t(s_t) = \log \beta_t(s_t)$$

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

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

variants:

discounted SOC: $Q_t(s_t, a_t) = r(s_t, a_t) + \gamma E[V_{t+1}(s_{t+1})]$ 

explicit temperature: $V_t(s_t) = \alpha \log \int \exp \left( \frac{1}{\alpha} Q_t(s_t, a_t) \right) da_t$
Policy computation summary

\[ \pi(a_t|s_t) = \exp(Q_t(s_t, a_t) - V_t(s_t)) = \exp(A_t(s_t, a_t)) \]

with temperature: \[ \pi(a_t|s_t) = \exp\left(\frac{1}{\alpha}Q_t(s_t, a_t) - \frac{1}{\alpha}V_t(s_t)\right) = \exp\left(\frac{1}{\alpha}A_t(s_t, a_t)\right) \]

- Natural interpretation: better actions are more probable
- Random tie-breaking
- Analogous to Boltzmann exploration
- Approaches greedy policy as temperature decreases
Forward messages

\[ p(O_t|s_t, a_t) \propto \exp(r(s_t, a_t)) \]

\[ p(s_{t+1}|s_t, a_t) \]

\[ \alpha_t(s_t) = p(s_t|O_{1:t-1}) \]

\[ = \int p(s_t|s_{t-1}, a_{t-1})p(a_{t-1}|s_{t-1}, O_{t-1})p(s_{t-1}|O_{1:t-2})ds_{t-1}da_{t-1} \]

\[ \alpha_{t-1}(s_{t-1}) \]

\[ p(a_{t-1}|s_{t-1}, O_{t-1}) = \frac{p(O_{t-1}|s_{t-1}, a_{t-1})p(a_{t-1}|s_{t-1})}{p(O_{t-1}|s_{t-1})} \]

\[ \beta_t(s_t) \]

what if we want \( p(s_t|O_{1:T})? \)

\[ p(s_t|O_{1:T}) = \frac{p(s_t, O_{1:T})}{p(O_{1:T})} = \frac{p(O_{t:T}|s_t)p(s_t, O_{1:t-1})}{p(O_{1:T})} \propto \beta_t(s_t)p(s_t|O_{1:t-1})p(O_{1:t-1}) \beta_t(s_t)\alpha_t(s_t) \]

\[ \alpha_t(s_t) \]
Forward/backward message intersection

$p(s_t) \propto \beta_t(s_t)\alpha_t(s_t)$

states with high probability of reaching goal

states with high probability of being reached from initial state (with high reward)

state marginals
Forward/backward message intersection

Li & Todorov, 2006

states with high probability of reaching goal

\[ p(s_t) \propto \beta_t(s_t)\alpha_t(s_t) \]

states with high probability of being reached from initial state (with high reward)

state marginals
Summary

1. Probabilistic graphical model for optimal control

2. Control = inference (similar to HMM, EKF, etc.)

3. Very similar to dynamic programming, value iteration, etc. (but “soft”)
Q-learning with soft optimality

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

target value: $V(s') = \max_{a'} Q_\phi(s', a')$

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

target value: $V(s') = \text{soft max}_{a'} Q_\phi(s', a') = \log \int \exp(Q_\phi(s', a')) da'$

$\pi(a|s) = \exp(Q_\phi(s, a) - V(s)) = \exp(A(s, a))$

1. take some action $a_i$ and observe $(s_i, a_i, s'_i, r_i)$, add it to $\mathcal{R}$
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 $Q_{\phi'}$
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 $\phi'$: copy $\phi$ every $N$ steps, or Polyak average $\phi' \leftarrow \tau \phi' + (1 - \tau)\phi$
**Policy gradient with soft optimality**

\[
\pi(a|s) = \exp(Q_\phi(s, a) - V(s)) \text{ optimizes } \sum_t E_{\pi(s_t, a_t)}[r(s_t, a_t)] + E_{\pi(s_t)}[\mathcal{H}(\pi(a|s_t))]
\]

policy entropy

**intuition:** $\pi(a|s) \propto \exp(Q_\phi(s, a))$ when $\pi$ minimizes $D_{KL}(\pi(a|s)\|\frac{1}{Z} \exp(Q(s, a)))$

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

often referred to as “entropy regularized” policy gradient
combats premature entropy collapse
turns out to be closely related to soft Q-learning:
see Haarnoja et al. ‘17 and Schulman et al. ‘17

Ziebart et al. ‘10 “Modeling Interaction via the Principle of Maximum Causal Entropy”
Policy gradient vs Q-learning

policy gradient derivation:

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

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

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

recall: \( \log \pi(a_t|s_t) = Q(s_t,a_t) - V(s_t) \)

\[ \approx \frac{1}{N} \sum_i \sum_t (\nabla_\theta Q(a_t|s_t) - \nabla_\theta V(s_t)) \left( r(s_t,a_t) + Q(s_{t+1},a_{t+1}) - Q(s_t,a_t) + V(s_t) \right) \]

Q-learning \( \frac{1}{N} \sum_i \sum_t \nabla_\theta Q(a_t|s_t) \left( r(s_t,a_t) + \text{soft max}_{a_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t) \right) \)

descent (vs ascent)
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 to hard optimality as reward magnitude increases
• Good model for modeling human behavior (more on this later)
Review

• Reinforcement learning can be viewed as inference in a graphical model
  • Value function is a backward message
  • Maximize reward and entropy (the bigger the rewards, the less entropy matters)
• Soft Q-learning
• Entropy-regularized policy gradient

\[ Q_\phi(s, a) \leftarrow r(s, a) + \gamma \text{soft } \max_{a'} Q_\phi(s', a') \]

\[ a = \arg \max_a Q_\phi(s, a) \]

fit a model to estimate return

generate samples (i.e. run the policy)

improve the policy
Stochastic models for learning control

- How can we track both hypotheses?
Stochastic energy-based policies

$Q$-function: $Q(s, a) : S \times A \to \mathbb{R}$

$$Q(s, a)$$

$$\pi(a|s) \propto \exp(Q(s, a))$$

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

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

$$V_t(s_t) = \log \int \exp(Q_t(s_t, a_t))a_t$$
Soft Q-learning

Learned (neural network) Q-function: $Q_\theta(s, a)$

Q-learning: $\theta \leftarrow \theta + \alpha \nabla_\theta Q_\theta(s, a)(r(s, a) + \gamma V(s') - Q_\theta(s, a))$

target value: $V(s') = \max_{a'} Q_\theta(s', a')$

soft Q-learning: $\theta \leftarrow \theta + \alpha \nabla_\theta Q_\theta(s, a)(r(s, a) + \gamma V(s') - Q_\theta(s, a))$

target value: $V(s') = \text{soft} \max_{a'} Q_\theta(s', a') = \log \int \exp(Q_\theta(s', a')) da'$
Tractable amortized inference for continuous actions

\[ \pi(a|s) \propto \exp(Q(s, a)) \]

Stochastic network:

\[ \xi \xrightarrow{s} \rightarrow a \]

Trained with amortized SVGD to match \( \pi(a|s) \propto \exp(Q(s, a)) \)
Stochastic energy-based policies provide pretraining
More work on maximum entropy policies


O’Donoghue et al. Combining Policy Gradient and Q-Learning. 2017

Soft optimality suggested readings

• Todorov. (2008). General duality between optimal control and estimation: primer on the equivalence between inference and control.