CS 4789 Final Review

Contents

1 MDP Definitions

- S states, A actions
- r map from state, action to scalar reward
- P transition probability to next state given current state and action (Markov assumption)
- γ discount factor
- H horizon
- μ_0 initial distribution

2 Optimal Control Problem

- Continuous states/actions $S \sim \mathbb{R}^{n_s}, A \sim \mathbb{R}^{n_a}$
- Cost instead of reward
- Transitions P described in terms of dynamics function and disturbance $w \sim D$: $s' = f(s, a, w)$

3 Policies and Distributions

- Policy π chooses an action based on the current state so $a_t = a$ with probability $\pi(a|s_t)$
- Shorthand for deterministic policy: $a_t = \pi(s_t)$
- Probability for trajectory $\tau = (s_0, a_0, \ldots, s_t, a_t)$

$$\mathbb{P}^{\pi}_{\mu_0}(\tau) = \mu_0(s_0)\pi(a_0|s_0) \cdot \Pi_{i=1}^{t} P(s_i|s_{i-1}, a_{i-1})\pi(s_i|s_i)$$

- Probability of (s,a) at t

$$\mathbb{P}^{\pi}_{t}(s, a; \mu_0) = \sum_{s_{0:t-1}\ a_{0:t-1}} \mathbb{P}^{\pi}_{\mu_0}(s_{0:t-1}, a_{0:t-1}, s_t, a_t | s_t = s, a_t = a)$$

- Discounted "steady-state" distribution

$$d^{\pi}_{\mu_0}(s, a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}^{\pi}_{t}(s, a; \mu_0)$$

$$V^{\pi}(\mu_0) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s,a \sim d^{\pi}_{\mu_0}} [r(s, a)]$$

- Finite horizon "steady-state" distribution

$$d^{\pi}_{\mu_0}(s, a) = \frac{1}{H}\sum_{t=0}^{H-1} \mathbb{P}^{\pi}_{t}(s, a; \mu_0)$$

4 Value and Q function

- Discounted Infinite Horizon

$$V^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s]$$

$$Q^\pi(s,a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a]$$

- Finite Horizon

$$V_t^\pi(s) = \mathbb{E}[\sum_{k=t}^{H-1} r_k | s_t = s]$$

$$Q_t^\pi(s,a) = \mathbb{E}[\sum_{k=t}^{H-1} r_k | s_t = s, a_t = a]$$

Recursive Bellman Expectation Equation:

- Discounted Infinite Horizon

$$V^\pi(s) = \mathop{\mathbb{E}}_{a \sim \pi(s)}[r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(s,a)}[V^\pi(s')]]$$

$$Q^\pi(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(s,a)}[V^\pi(s')]$$

  - Recursive computation:
  $$V^\pi = R^\pi + \gamma P^\pi V^\pi$$
  - Exact Policy Evaluation:
  $$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$$
  - Iterative Policy Evaluation:
  $$V_{t+1}^\pi = R^\pi + \gamma P^\pi V_t^\pi$$

- Finite Horizon

$$V_t^\pi(s) = \mathop{\mathbb{E}}_{a \sim \pi(s)}[r(s,a) + \mathop{\mathbb{E}}_{s' \sim P(s,a)}[V^\pi(s')]]$$

$$Q^\pi(s,a) = r(s,a) + \mathop{\mathbb{E}}_{s' \sim P(s,a)}[V^\pi(s')]$$

  - Backwards iterative computation in finite horizon
    Initialize $V_H^\pi = 0$
    For $t = H-1, H-2, \ldots, 0$:

  $$V_t^\pi = R^\pi + P^\pi V_{t+1}^\pi$$

5 Optimal Policies

An optimal policy π^* is one where $V^{\pi^*}(s) \geq V^\pi(s)$ for all s and policies π

5.1 Infinite Horizon

- Equivalent condition: Bellman Optimality

$$V^*(s) = \max_{a \in A}[r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(s,a)}[V^*(s')]$$

$$Q^*(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(s,a)}[\max_{a' \sim A} Q^*(s',a')]$$

- Optimal Policy: $\forall s \in S$,

$$\pi^*(s) = \operatorname*{argmax}_{a \sim A} Q^*(s,a)$$

- In infinite horizon, we use value iteration/policy iteration
- Value Iteration

  Initialize Q_0

  For $t = 0, 1, \ldots$:

$$Q^{t+1} = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(s,a)}[\max_{a' \sim A} Q^t(s',a')]$$

- Policy Iteration

  Initialize π_0

  For $t = 0, 1, \ldots$:

$$Q^t = \text{ PolicyEval } (\pi^t)$$

$$\pi_{t+1} = \operatorname*{argmax}_{a \in A} Q^t(s,a)$$

5.2 Finite Horizon

- Solve by dynamic programming: iterate backwards in time from $V_H^* = 0$
- Initialize $V_H^*(s) = 0$

  For $t = 0, \ldots, H-1$

$$V^*(s) = \max_{a \in A}[r(s,a) + \mathop{\mathbb{E}}_{s' \sim P(s,a)}[V_{t+1}^*(s')]$$

$$Q_t^*(s,a) = r(s,a) + \mathop{\mathbb{E}}_{s' \sim P(s,a)}[\max_{a' \sim A} Q_{t+1}^*(s',a')]$$

- Optimal Policy: $\forall s \in S$,

$$\pi_t^*(s) = \operatorname*{argmax}_{a \sim A} Q_t^*(s,a)$$

6 Linear Optimal Control

- Linear Dynamics:
$$s_{t+1} = As_t + Ba_t + w_t,\ w_t \sim \mathcal{N}(0, \sigma^2 I)$$
- Unrolled Dynamics
$$s_t = A^t s_0 + \sum_{k=0}^{t-1} A^k (Ba_{t-k-1} + w_{t-k-1})$$
- Stability of uncontrolled $s_{t+1} = As_t$: determined by whether $\rho(A) < 1$
- Finite Horizon LQR: Application of dynamic programming and local linearization

7 Learning From Data

What do we want to learn?

- Unknown transitions $P(s'|s, a)$
- Reward function $r(s, a)$
- Value/Q function of policy or optimal policy
- Optimal Policy $\pi^*(s)$

Fitting a model:

- Via counting:
$$\hat{f}(x) = \sum_{i=1}^{N} y_i \frac{\mathbb{1}\{x = x_i\}}{\sum_{i=1}^{N}} \mathbb{1}\{x = x_i\}$$
- Function approximation:
$$\hat{f}(x) = \min_{f \in F} \frac{1}{N} \sum_{i=1}^{N} (f(x) - y)^2$$

8 Learning Models/Model-Based RL

Meta-Algorithm: Model Based RL

1. For $i = i, \ldots, N$:
   Sample $s'_i \sim P(s_i, a_i)$ and reward $r(s'_i, s_i, a_i)$
2. Fit transition model $\hat{P}$ from data $\{(s'_i, s_i, a_i)\}_{i=1}^{N}$
3. Design $\hat{\pi}$ using $\hat{P}$

Tabular setting: $\hat{P}$ via counting

1. Sample all (s,a) evenly: $\frac{N}{SA}$ times each
2. Fit transition model by counting

$$\hat{P}(s'|s,a) = \frac{\sum_{i=1}^{N} \mathbb{1}\{s_i = s \ \&\ a_i = a\}\mathbb{1}\{s_i' = s'\}}{\sum_{i=1}^{N} \mathbb{1}\{s_i = s \ \&\ a_i = a\}}$$

3. Design $\hat{\pi}$ with policy iteration $PI(\hat{P}, r)$:
   Initialize π^0
   For $t = 1, \ldots, T$:

$$Q^{\pi^t} = \text{PolicyEval}(\pi^t; \hat{P}, r)$$
$$\pi^t(s) = \operatorname*{argmax}_a Q^{\pi^t}(s,a)$$

Simulation Lemma: translate $\hat{P}$ v.s. P into $\hat{V}$ v.s. V

$$\hat{V}^\pi(s_0) - V^\pi(s_0) \leq \frac{\gamma}{(1-\gamma)^2} \mathop{\mathbb{E}}_{s,a\sim d^\pi_{s_0}} [\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\|_1]$$

Features are (s_i, a_i)

$$(s_i, a_i) = (s_{h_1}, a_{h_1}) \sim d^\pi_{\mu_0},\ h_1 = h \text{ with probability} \propto \gamma^h$$

Labels constructed as:

- Rollout based (MCMC):

$$y_i = \sum_{h_1}^{h_1+h_2} r_t$$

  Advantage: unbiased estimator
  Disadvantage: high variance

- Bellman Expectation based:

$$y_t = r_t + \gamma \hat{Q}(s_{t+1}, a_{t+1})$$

- Bellman Optimality based (TD):

$$y_t = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a)$$

  Advantage: low variance
  Disadvantage: biased estimator

$$\hat{Q} = \min_{Q\in\mathcal{Q}} \sum_{i=1}^{N} [Q(s_i,a_i) - y_i]^2$$

8.1 Rollout with breaks

Initialize $s_0 = s, a_0 = a$

For $t = 0, 1, \ldots$:

- Take action a_t and observe $r_t = r(s_t, a_t), s_{t+1} \sim P(s_t, a_t)$
- With probability $1 - \gamma$:
  Break and return $y = \sum_{k=0}^{t} r_t$
- Update $a_{t+1} = \pi(s_{t+1})$

8.2 Sample

Initialize $s_0 \sim \mu_0, a_0 = \pi(s_0)$

For $t = 0, 1, \ldots$

- Take action a_t and observe $S_{t+1} \sim P(s_t, a_t)$
- With probability $1 - \gamma$:
  break and return a_t, s_t
- Update $a_{t+1} = \pi(s_{t+1})$

This algorithm is equivalent to sampling from $d^{\pi}_{\mu_0}$ (on policy).

8.3 ROLLOUTAPPROX

For $i = 1, \ldots, N$

- s_i, a_i = Sample(π)
- y_i = ROLLOUTWITHBREAKS(s_i, a_i, π)

$\hat{Q}^{\pi} = \text{argmin}_{Q \in Q} \sum_{i=1}^{N} (Q(s_i, a_i) - y_i)^2 \leftarrow$ empirical risk minimization with squared loss

8.4 Approximated Dynamic Programming

Initialize π_0

For $t = 0, 1, \ldots$:

1. $\hat{Q}^t$ = SampleAndEvaluate(π^t)
2. π^{t+1} = Improvement($\hat{Q}_t$)

8.5 Approximate Policy Iteration

For $t = 0, 1, \ldots$:

- $\hat{Q}^{\pi^t} = \text{ROLLOUTAPPROX}(\pi_t) \leftarrow$ regression-based
- $\pi_{t+1}(s) = \text{argmax}_a \hat{Q}^{\pi_t}(s, a) \leftarrow$ policy improvement same as PI

Greedy improvement, could oscillate.

8.6 Conservative Policy Iteration

for $t = 0, 1, \ldots$

- $\hat{Q}^{\pi^t} = \text{ROLLOUTAPPROX}(\pi_t)$
- $\pi'(s) = \text{argmax}_a \hat{Q}^{\pi_t}(s, a)$
- $\pi_{t+1}(s) = (1 - \alpha)\pi_t(\cdot|s) + \alpha\pi'(\cdot|s) \leftarrow$ incremental update controlled by stepsize $\alpha \in [0, 1]$

Incremental improvement.

8.7 Performance Difference Lemma

Goal: understand V^π v.s. $V^{\pi'}$ in terms of the difference between π v.s. π'

$$V^\pi(s_0) - V^{\pi'}(s_0) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s \sim d^\pi_{s_0}} \left[\mathop{\mathbb{E}}_{a \sim \pi(s)} [Q^{\pi'}(s, a)] - V^{\pi'}(s) \right]$$

$$\mathop{\mathbb{E}}_{s \sim \mu} \left[V^\pi(s) - V^{\pi'}(s) \right] = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s \sim d^\pi_\mu} \left[\mathop{\mathbb{E}}_{a \sim \pi(s)} [A^{\pi'}(s, a)] \right]$$

$$|V^\pi(s_0) - V^{\pi'}(s_0)| \le \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s \sim d^\pi_{s_0}} \left[\sum_{a \in A} \left| \pi(a|s) - \pi'(a|s) \right| Q^{\pi'}(s, a) \right]$$

8.8 SARSA/State-Action-Reward-State-Action (Supervision via Bellman Equation)

Initialize $Q_0, s_0 \sim \mu_0, a_0 \sim \pi(s_0)$
for $t = 0, 1, \ldots$

- Take action a_t and observe $S_{t+1} \sim P(s_t, a_t)$ and $r_t \sim r(s_t, a_t)$
- Sample $a_{t+1} \sim \pi(s_{t+1})$
- Update $Q^{t+1}(s_t, a_t) = (1 - \alpha)Q^t(s_t, a_t) + \alpha(r_t + \gamma Q^t(s_{t+1}, a_{t+1}))$

Fixed point iteration. $\hat{Q}$ will approach true Q^π (also on policy). Biased label when $\hat{Q} \neq Q^\pi$.

8.9 Policy Improvement with epsilon-greedy

SARSA requires sufficient exploration to converge.

A common strategy is $\epsilon-$greedy:

$$\pi(s) = \begin{cases} \text{argmax}_a Q(s,a) \text{ w.p, } 1-\epsilon \\ a_0 \text{ w.p, } \frac{\epsilon}{A} \\ a_1 \text{ w.p, } \frac{\epsilon}{A} \\ \dots \end{cases}$$

Equivalently,

$$\pi(a|s) = \begin{cases} 1-\epsilon+\frac{\epsilon}{A}, \ a = \text{argmax}_a Q(s,a) \\ \frac{\epsilon}{A}, \ \text{otherwise} \end{cases}$$

8.10 Supervision via Bellman Optimality

Recall Bellman Optimality:

$$Q^*(s,a) = r(s,a) + \gamma\mathbb{E}[max_{a'} Q^*(s',a')]$$

Initialize Q

for $t = 0, 1, \dots$

- Take action a_t (e.g. $\epsilon - greedy$) and $s_{t+1} \sim P(s_t, a_t), r_t \sim r(s_t, a_t)$
- $Q(s,a) = (1-\alpha)Q(s,a) + \alpha(r_t + \gamma \max_{a'} Q(s_{t+1}, a'))$

Properties of Bellman Optimality based supervision:

1. Updates at every timestep.
2. Biased label when $Q \neq Q^*$.
3. Variance depends on randomness from one timestep.
4. Not specific to a policy, so can use off-policy data.

9 Policy Optimization

- $J(\theta)$ =expected cumulative reward under policy π_θ
- Estimate $\nabla_\theta J(\theta)$ via rollouts τ, observed rewards $R(\tau)$
  - Random Search:

$$\theta \pm \delta v, \ g = \frac{1}{2\delta}(R(\tau_+) - R(\tau_-))v$$

- REINFORCE (Policy gradient from trajectories)
  An unbiased estimate of $\nabla J(\theta)$:

$$g = \sum_{t=0}^{\infty} \nabla_\theta[\log(\pi_\theta(a_t|s_t))]R(\tau)$$

- Actor-Critic
  An unbiased estimate of $\nabla J(\theta)$:

$$g = \frac{1}{1-\gamma}[\nabla_\theta \log(\pi_\theta(a|s))]Q^{\pi_\theta}(s,a)$$

  Final gradient estimate (baseline function $b(s)$ reduces variance):

$$g = \frac{1}{1-\gamma}[\nabla_\theta \log(\pi_\theta(a|s))][Q^{\pi_\theta}(s,a) - b(s)]$$

  For any action-independent baseline $b(s)$:

$$\mathop{\mathbb{E}}_{a\sim\pi_\theta(s)}[\nabla_\theta \log(\pi_\theta(a|s))b(s)] = 0$$

Meta-Algorithm: Derivative-Free SGA

Initialize θ_0

For $t = 0, 1, \ldots$:

1. Collect rollouts using θ_t
2. Compute (estimate) $g_t = \nabla_{\theta_t} J(\theta_t)$
3. $\theta_{t+1} = \theta_t + \alpha g_t$

- Trust regions and Natural Policy Gradient

$$\max \nabla_\theta J(\theta)^\top(\theta - \theta_0) \text{ such that } (\theta-\theta_0)^\top F_{\theta_0}(\theta-\theta_0) \leq \delta$$

$$\theta_{t+1} = \alpha F^{-1} g_t$$

- K-L Divergence: measures the "distance" between two distributions.

$$KL(P|Q) = \mathop{\mathbb{E}}_{x\sim P}[\log \frac{P(x)}{Q(x)}] = \sum_{x\in X} P(x)\log(\frac{P(x)}{Q(x)})$$

  Facts:

  - $KL(P|Q) \geq 0$
  - $KL(P|Q) = 0 \iff P = Q$
  - Not necessarily symmetric (treat Q as baseline, compared to P)

$$d_{KL}(\theta_0, \theta) = \mathop{\mathbb{E}}_{s,a\sim d_{\mu_0}^{\pi_{\theta_0}}} \log \frac{\pi_{\theta_0}(a|s)}{\pi_\theta(a|s)}$$

10 Exploration

Regret:

$$R(T) = \sum_{t=1}^{T} \mathbb{E}[\mu^*(x_t) - \mu_{a_t}(x_t)]$$

Goal: sublinear regret. If $R(T)$ is sublinear, then

$$\lim_{T \to \infty} \frac{R(T)}{T} \to 0$$

Both "Random" (pure explore) and "Greedy" (pure exploit) approach suffer from linear regret.

10.1 Explore-then-commit

For $t = 1, \ldots, NK$

$a_t = t \mod k$

$\hat{\mu}_a = \frac{1}{N} \sum_{i=1}^{N} r_{k \cdot i}$

For $t = NK + 1, \ldots, T$

$a_t = \text{argmax}_a \hat{\mu}_a = \hat{a}^*$

The regret can be decomposed into:

$$R(T) = \sum_{t=1}^{T} \mu^* - \mu_{a_t} = \sum_{t=1}^{NK} \mu^* - \mu_{a_t} + \sum_{t=NK+1}^{T} \mu^* - \mu_{\hat{a}^*} = R_1 + R_2$$

Lemma: After exploration phase, for all arms $a = 1, \ldots, k$,

$$|\hat{\mu}_a - \mu_a| \leq \sqrt{\frac{\log(\frac{k}{\delta})}{N}} \text{ with probability } 1 - \delta$$

Proof: Hoeffding Bound and Union Bound

Lemma (Hoeffding): Suppose $r_i \in [0, 1]$ and $\mathbb{E}[r_i] = \mu$. Then for $r_1, \ldots, r_N$ i.i.d with probability $1 - \delta$,

$$|\hat{\mu} - \mu| = |\frac{1}{N} \sum_{i=1}^{N} r_i - \mu| \leq \sqrt{\frac{\log(\frac{1}{\delta})}{N}}$$

If with probability $\frac{\delta}{k}$,

$$|\hat{\mu} - \mu| \geq \sqrt{\frac{\log(\frac{k}{\delta})}{N}},$$

then by union bound,

$$Pr(\text{There exists an arm a = 1,...,k such that } \hat{\mu}_a - \mu_a \geq \sqrt{\frac{\log(\frac{k}{\delta})}{N}}) \leq k \cdot \frac{\delta}{k} = \delta$$

Therefore,

$$Pr(\text{For all arms a = 1,...,k, } \hat{\mu}_a - \mu_a \leq \sqrt{\frac{\log(\frac{k}{\delta})}{N}}) \geq 1 - \delta$$

To bound $R_2 = \sum_{t=NK+1}^{T} \mu^* - \mu_{\hat{a}^*}$, apply the above lemma,

$$\mu_a \in [\hat{\mu}_a \pm \sqrt{\frac{\log(\frac{k}{\delta})}{N}}]$$

$$\begin{aligned}
R_2 &= \sum_{t=NK+1}^{T} \mu^* - \mu_{\hat{a}^*} \\
&= (T - NK)(\mu^* - \mu_{\hat{a}^*}) \\
&\leq (T - NK)[(\hat{\mu}_{a^*} + \sqrt{\frac{\log(\frac{k}{\delta})}{N}}) - (\hat{\mu}_{\hat{a}^*} - \sqrt{\frac{\log(\frac{k}{\delta})}{N}})] \\
&\leq (T - NK)(2\sqrt{\frac{\log(\frac{k}{\delta})}{N}}) \text{ since } \hat{\mu}_{a^*} - \hat{\mu}_{\hat{a}^*} \leq 0 \text{ by definition of } \hat{a}^*
\end{aligned}$$

Finally, we have

$$R(T) = R_1 + R_2 \leq NK + 2T\sqrt{\frac{\log(\frac{k}{\delta})}{N}} \text{ with probability } 1 - \delta$$

Minimize this upper bound with respect to N (take derivative and set to zero),

$$N = \left(\frac{T}{2k}\sqrt{\log(\frac{k}{\delta})}\right)^{\frac{2}{3}}$$

$$R(T) \leq T^{\frac{2}{3}} k^{\frac{1}{3}} [\log(\frac{k}{\delta})]^{\frac{1}{3}}$$

Regret is sublinear! $R(T) \sim O(T^{\frac{2}{3}})$

10.2 Upper Confidence Bound Algorithm (UCB)

Initialize $\hat{\mu}_0^a, N_0^a$ for $a = 1, \ldots, k$

For $t = 1, 2, \ldots, T$:

$a_t = \text{argmax}_a \, \hat{\mu}_t^a + \sqrt{\frac{\log(\frac{kT}{\delta})}{N_t^a}} = \text{argmax}_a \, \hat{u}_t^a$

Update $\hat{\mu}_{t+1}^{a_t}$ and $N_{t+1}^{a_t}$

Note that we have defined $\hat{u}_t^a = \mu_t^a + \sqrt{\frac{\log(\frac{kT}{\delta})}{N_t^a}}$ to be the UCB at time t.

The reason for $\frac{KT}{\delta}$ is $\frac{\delta}{KT} \cdot K \cdot T = \delta$.

This is like adding a synthetic reward bonus inversely proportional to the number of times we visit a state.

UCB Analysis:

Regret at time t:

$$\begin{aligned}
\mu^* - \mu_{a_t} &\leq \hat{u}_t^{a^*} - \mu_{a_t} \\
&\leq \hat{u}_t^{a_t} - \mu_{a_t} \\
&= \hat{\mu}_t^{a_t} + \sqrt{\frac{\log(\frac{kT}{\delta})}{N_t^{a_t}}} - \mu_{a_t} \\
&\leq 2\sqrt{\frac{\log(\frac{kT}{\delta})}{N_t^{a_t}}}
\end{aligned}$$

where we have used

$$\hat{\mu}_t^{a_t} - \mu_{a_t} \leq \sqrt{\frac{\log(\frac{kT}{\delta})}{N_t^{a_t}}}$$

Putting it all together,

$$\begin{aligned}
R(T) = \sum_{t=1}^{T} \mu^* - \mu_{a_t} &\leq 2\sqrt{\log(\frac{kT}{\delta})} \sum_{t=1}^{T} \sqrt{\frac{1}{N_t^{a_t}}} \\
&\leq 2\sqrt{\log(\frac{kT}{\delta})}\sqrt{kT} \\
&= 2\sqrt{kT\log(\frac{kT}{\delta})}
\end{aligned}$$

where we have used the fact that

$$\sum_{t=1}^{T} \sqrt{\frac{1}{N_t^{a_t}}} \leq \sqrt{kT}$$

Sublinear regret! $R(T) \sim O(\sqrt{T})$

10.3 Contextual Bandits

10.3.1 Motivation:

In reality, contexts include many pieces of information, and the number of discrete contexts may be very large! We may never the exact context twice!
Correlation exist between similar contexts.

10.3.2 Explote-then-commit with functional approximation

1. Pull each arm N times and record $\left\{\{x_i^a, r_i^a\}_{i=1}^N\right\}_{a=1}^k$
   Estimate $\hat{\mu}_a(x) = \text{argmin}_{\mu \in \mathcal{M}} \sum_{i=1}^N \left(\mu(x_i^a) - r_i^a\right)^2$
2. For $t = NK+1, \ldots, T$, pull $a)t = \text{argmax}_a \hat{\mu}_a(x_t)$

Lemma: For $x_i \sim D$ i.i.d and $\mathbb{E}[y_i] = f_*(x_i)$ for some $f_* \in \mathcal{F}$, and

$$\hat{f} = \underset{f \in \mathcal{F}}{\text{argmin}} \sum_{i=1}^{N} (\hat{f}(x_i) - y_i)^2$$

then with high probability,

$$\mathop{\mathbb{E}}_{x\sim D}[|\hat{f}(x) - f_*(x)|] \leq \sqrt{\frac{C_{\mathcal{F}}}{N}}$$

10.3.3 Linear Contextual Bandits

Setting: simplified MDP consists of

- Contexts $x \in X \subseteq \mathbb{R}^d$
- Actions "arms" $a \in A = \{1, \ldots, k\}$
- Rewards $r_t = r(x_t, a_t)$ with $\mathbb{E}[r(x,a)] = \mu_a(x) = \theta_a^\top x$
- Horizon T

Goal: find a policy $a_t = \pi(x_t)$ that achieves low regret.

$$R(T) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{x_t\sim D}[\max_a \theta_a^\top x_t - \theta_{a_t}^\top x_t]$$

The problem reduces to finding

$$\hat{\theta}_a = \operatorname*{argmin}_\theta \sum_{i=1}^{N} (\theta^\top x_i^a - r_i^a)^2$$

Lemma: (by taking the gradient and set to zero) as long as $(x_i)_{i=1}^N$ span $\mathbb{R}^d$,

$$\hat{\theta} = (\sum_{i=1}^{N}) x_i x_i^\top)^{-1} \sum_{i=1}^{N} x_i r_i = A^{-1} b$$

The matrix A is related to the empirical covariance

$$\Sigma = \mathop{\mathbb{E}}_{x\sim D}[xx^\top]$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T = \frac{A}{N}$$

10.3.4 Upper Confidence Bound Value Iteration

Initialize transition probability $\hat{P}_0$, reward bonus $b_0(s,a)$
For $i = 0, \ldots, T$:

- Optimistically plan: $\pi^i = VI(\hat{P}_i, r + b_i)$
- Collect new trajectory with π^i
- Update $\hat{P}_{i+1}$ and b_{i+1}

Reward bonus: encourage exploration of new state-action pairs

$$b_i(s,a) = H\sqrt{\frac{\alpha}{N_i(s,a)}}$$

Generate policy: VI reduces to dynamic programming!
Initialize $\hat{V}^i_H(s) = 0$
For $t = H-1, H-2, \ldots, 0$:

- $\hat{Q}^i_t = r(s,a) + b_i(s,a) + \mathop{\mathbb{E}}_{s' \sim \hat{P}(sma)}[\hat{V}^i_{t+1}(s')]$
- $\pi^i_t(s) = \text{argmax}_a\, \hat{Q}^i_t(s,a)$
- $\hat{V}^i_t(s) = \hat{Q}^i_t(s, \pi^i_t(s))$

Lemma: (optimism) as long as $r(s,a) \in [0,1]$,

$$\hat{V}^i_t(s) \geq V^*_t(s) \forall t, i, s$$
$$\hat{Q}^i_t(s,a) \geq Q^*_t(s,a) \forall t, i, s, a$$

11 Learning From Experts

11.1 Behavior Cloning

Expert knows optimal policy π^* and we have a dataset

$$D = \{s^*_i, a^*_i\}_{i=1}^M \sim d^{\pi^*}$$

Estimate a policy with empirical risk minimization

$$\hat{\pi} = \mathop{\text{argmax}}_{\pi \in \Pi} \sum_{i=1}^{N} l(\pi, s^*_i, a^*_i)$$

Analysis see lecture 22.

11.2 DAgger

Initialize π_0 and dataset $D = \emptyset$

For $t = 0, \ldots, T-1$:

1. Generate dataset with π_t and query expert
   $D^t = \{s_i, a^*_i\}$ where $s_i \sim d^{\pi^t}_\mu$ and $a^*_i = \pi^*(s_i)$
2. Data aggregation: $D = D \cup D^t$
3. Update policy via supervised learning

$$\pi^{t+1} = \mathop{\text{argmin}}_{\pi \in \Pi} \sum_{s,a \in D} l(\pi, s, a)$$

Analysis see lecture 23.

11.3 Max-Entropy Inverse RL

11.3.1 Entropy

Definition:

$$Ent(P) = \mathop{\mathbb{E}}_{x \sim P}[-\log(P(x))] = -\sum_{x \in X} P(x) \log(P(x))$$

Entropy = 0 only when distribution is deterministic. Otherwise, entropy is positive.

11.4 Lagrange Formulation

Initialize w_0

For $t = 0, \ldots, T-1$:

- $x_t = \text{argmin}_x f(x) + w_t g(x)$ [Best response]
- $w_{t+1} = w_t + \eta g(x_t)$ [Iterative update]

Return $\bar{x} = \frac{1}{T} \sum_{t=0}^{T-1} x_t$

11.5 Algorithm

Key assumption:

$$r(s, a) = \theta_*^\top \phi(s, a)$$

Linear reward with respect to features. $\theta_*^\top$ is unknown, $\phi(s, a)$ is known.

The Max Entropy RL method:

$$\max_\pi Ent(\pi) \text{ such that } \mathop{\mathbb{E}}_{d_\mu^{\pi^*}}[\phi(s, a)] = \mathop{\mathbb{E}}_{d_\mu^\pi}[\phi(s, a)]$$

Among consistent policies with the expert, and choose the one with the most uncertainty.

We can write out the constraint

$$g(w) = \mathop{\mathbb{E}}_{d_\mu^{\pi^*}}[\phi(s, a)] - \mathop{\mathbb{E}}_{d_\mu^\pi}[\phi(s, a)]$$

Goal:

$$\min_\pi \max_{w \in \mathbb{R}^d} \mathop{\mathbb{E}}_{s,a \sim d_\mu^\pi}[\log \pi(a|s)] + w^\top \left(\mathop{\mathbb{E}}_{s,a \sim d_\mu^{\pi^*}} \phi(s, a) - \mathop{\mathbb{E}}_{s,a \sim d_\mu^\pi} \phi(s, a) \right) = \min_\pi \max_{w \in \mathbb{R}^d} \mathcal{L}(\pi, w)$$

where we have defined

$$\mathcal{L}(\pi, w) = \mathop{\mathbb{E}}_{s,a \sim d_\mu^\pi} \left[\log \pi(a|s) - w^\top \phi(s, a) \right] + w^\top \mathop{\mathbb{E}}_{s,a \sim d_\mu^{\pi^*}} \phi(s, a)$$

11.5.1 Iterative Max-Ent IRL

Initialize $w_0 \in \mathbb{R}^d$

For $t = 0, \ldots, T-1$:

$$\pi_t = \underset{\pi}{\operatorname{argmax}} \underset{s,a \sim d_\mu^\pi}{\mathbb{E}} \Big[-\log \pi(a|s) + w_t^\top \phi(s,a) \Big]$$

$$w_{t+1} = w_t + \eta \Big(\underset{s,a \sim d_\mu^{\pi^*}}{\mathbb{E}} \phi(s,a) - \underset{s,a \sim d_\mu^\pi}{\mathbb{E}} \phi(s,a) \Big)$$

Return $\bar{\pi} = \frac{1}{T} \sum_{t=0}^{T-1} \pi_t$ We can view $w_t^\top \phi(s,a)$ as reward $r(s,a)$.

11.5.2 Soft Value Iteration

Use dynamic programming:

$$\underset{\pi}{\operatorname{argmax}} \underset{s,a \sim d_\mu^\pi}{\mathbb{E}} \Big[\sum_{t=0}^{H-1} r(s_t, a_t) - \log \pi_t(a_t|s_t) \Big| s_{t+1} \sim P(s_t, a_t), a_t \sim \pi_t(s_t), s_0 \sim \mu \Big]$$

Initialize $V_H^*(s) = 0$

For $h = H-1, \ldots, 0$:

$$Q_h^*(s,a) = r(s,a) + \underset{s' \sim P(s,a)}{\mathbb{E}} [V_{h+1}(s')]$$

$$\pi_h^*(a|s) \propto e^{Q_h^*(s,a)}$$

$$V_h^*(s) = \log \Big(\sum_{a \in A} e^{Q_h^*(s,a)} \Big)$$

Derivation of $\pi_h^*(\cdot|s)$:

$$\begin{aligned}
\pi_h^*(\cdot|s) &= \underset{\pi}{\operatorname{argmax}} \underset{a \sim \pi(\cdot|s)}{\mathbb{E}} \Big[Q_h^*(s,a) - \log \pi(a|s) \Big] \\
&= \underset{\rho \in \Delta(A)}{\operatorname{argmax}} \sum_{a \in A} \rho(a) \Big[Q_h^*(s,a) - \log \rho(a) \Big] \text{ such that } \sum_{a \in A} \rho(a) = 1 \\
&= \frac{e^{Q_h^*(s,a)}}{\sum_{a' \in A} e^{Q_h^*(s,a')}}
\end{aligned}$$

Derivation of $\rho(a)$:

$$\mathcal{L}(\rho, w) = \sum_{a \in A} \rho(a) Q_h^*(s,a) - \rho(a) \log(\rho(a)) + w \Big([\sum_{a \in A} \rho(a)] - 1 \Big)$$

$$\frac{\partial \mathcal{L}}{\partial \rho(a)} = Q_h^*(s,a) - \log \rho(a) - \frac{\rho(a)}{\rho(a)} + w = 0$$

$$\rho(a) = e^{Q_h^*(s,a)} e^{w-1}, \ \forall a$$

where e^{w-1} is the normalization factor $\left(\sum_{a \in A} \rho(a) = 1\right)$

$$\rho(a) = \frac{e^{Q_h^*(s,a)}}{\sum_{a' \in A} e^{Q_h^*(s,a')}}$$

Derivation of $V_h^*(s)$:

$$\begin{aligned} V_h^*(s) &= \mathop{\mathbb{E}}_{a \sim \pi_h^*(\cdot|s)}\left[Q_h^*(s,a) - \log \pi_h^*(a|s)\right] \\ &= \log\left(\sum_{a \in A} e^{Q_h^*(s,a)}\right) \end{aligned}$$

12 Proof Strategies

1. Add and subtract
$$\|f(x) - g(y)\| \le \|f(x) - f(y)\| - \|f(y) - g(y)\|$$
2. Contractions (induction)
$$\|x_{t+1} \le \gamma \|x_t\|\|\| \to \|x_t\| \le \gamma^t \|x_0\|$$
3. Additive induction
$$\|x_{t+1}\| \le \delta_t + \|x_t\| \to \|x_t\| \le \sum_{k=0}^{t-1} \delta_k + \|x_0\|$$
4. Basic inequalities
$$|\mathbb{E}[f(x)] - \mathbb{E}[g(x)]| \le \mathbb{E}[|f(x) - g(x)|]$$
$$|\max f(x) - \max g(x)| \le \max |f(x) - g(x)|$$
$$\mathbb{E}[f(x)] \le \max f(x)$$