Review

- MDP = \{ S, A, P(s' | s, a), R(s) \}

- State value function
  \[ V^\pi(s) = E^\pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s \right] \]

- Bellman equation
  \[ V^\pi(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi(s)) V^\pi(s') \]

- Action value function
  \[ Q^\pi(s, a) = R(s) + \gamma \sum_{s'} P(s' \mid s, a) V^\pi(s') \]

- Optimality
  \[ \exists \pi^* \text{ such that } V^{\pi^*}(s) \geq V^\pi(s) \text{ for all } \pi, s \]
  \[ V^*(s) \triangleq V^{\pi^*}(s) \]
  \[ Q^*(s, a) \triangleq Q^{\pi^*}(s, a) \]

Value iteration

- Bellman optimality equation
  \[ V^*(s) = R(s) + \gamma \max_a \sum_{s'} P(s' \mid s, a) V^*(s') \]

- Iteration:
  \[ V(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s' \mid s, a) V(s') \]
Thm: last lecture

\[ \Delta_k = \max_s |V_k(s) - V^*(s)| \]

iteration

\[ \Delta_k \leq \gamma^k \Delta_0 \]

Assume rewards are bounded:

\[ \Delta_0 = \max_s |V_0(s) - V^*(s)| \]

\[ = \max_s |V^*(s)| \]

\[ \Delta_0 \leq \left[ \max_s |R(s)| \right] (1 + \gamma + \gamma^2 + ...) \]

\[ = \left[ \max_s |R(s)| \right] \cdot \frac{1}{1-\gamma} \]

Thm: \( \Delta_k \leq \left( \left[ \frac{\gamma^k}{1-\gamma} \right] \cdot \max_s |R(s)| \right) \rightarrow 0 \) as \( k \rightarrow \infty \)

Convergence rate depends on \( \gamma \).

More iterations are required as \( \gamma \rightarrow 1 \).

---

**Reinforcement learning (RL)**

- What if \( P(s'|s,a) \) and \( R(s) \) are not known?
  - Can we learn \( \Pi^*(s) \) or \( V^*(s) \) from experience?

  Experience:

  \[ s_0, a_0 \rightarrow s_1, a_1 \rightarrow s_2 \rightarrow ... \]

- Model-based approach (indirect method)
  - Explore world
  - Estimate model \( P_{ML}(s'|s,a) \) from stream of experience.
    - Hope that \( P(s'|s,a) \approx P_{ML}(s'|s,a) \) as agent gains more experience.
    - Compute \( \Pi^* \) from \( P_{ML}(s'|s,a) \).

- Disadvantage
  - To store \( P_{ML}(s'|s,a) \) is \( O(n^2) \) for \( n \) states.
  - Only care about \( \Pi^*(s) \) or \( V^*(s) \) which are \( O(n) \).
  - Is it really necessary to estimate a model?
Advantage

Model $p(s'|s,a)$ is useful for task transfer, where rewards $R(s)$ or discount factor $\gamma$ changes but
dynamics stay the same.

Ex: $p(s'|s,a)$ robot navigation
$R(s)$—go to door, leave building, etc...

Beyond CSE ISO

- Extension #1: MDPs with undiscounted rewards
  - Suppose goal is to maximize (or evaluate)
    $$\pi^* = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} R(s_t)$$
    Assume here that $\pi^*$ does not depend on initial state $s_0$.
    Certain states have better transients than others:
    $\tilde{V}^\pi(s) = E^\pi \left[ \sum_t \left[ R(s_t) - \pi^* \right] \mid s_0 = s \right]$
    $\tilde{Q}^\pi(s,a) = E^\pi \left[ \sum_t \left[ R(s_t) - \pi^* \right] \mid s_0 = s, a_0 = a \right]$

- Extension #2: large state spaces
  - So far: implicit assumption that we can store
    $V^\pi(s)$ or $\pi(s)$ as lookup table
  - Ex: value iteration
    $$V_{k+1}(s) = R(s) + \gamma \max_a \sum_{s'} p(s'|s,a) V_k(s')$$
  - Function approximation in RL
    - Storing $V^\pi(s)$ is impossible for (say) backgammon with $10^{50}$ states
    - $V^\pi(s, \theta)$ must be parameterized in terms of parameter vector $\theta$
Extension #3: partially observable Markov decision process (POMDP)

- POMDPs are to MDPs as HMMs are to Markov models
  
  Ex: robot navigation
  
  states: xy location
  observations: sensors

- Model for POMDP
  
  Transitions $P(s_{t+1} | s_t, a_t)$
  Rewards $R(s_t)$
  Observation $P(o_t | s_t)$

Optimal behavior/policy in POMDP depends on entire history of observations.

Much harder problem.
Overview

- principles vs. heuristics
  - optimizations vs. rules-of-thumb
    1) common-sense reasoning (e.g. explaining away) from Bayes rule
    2) learning as maximum likelihood estimation
    3) planning as optimal decision making in MDP
- compact representations of complex worlds
  - power/expressiveness vs. tractability
    1) naive Bayes model for document classification
    2) Markov models of language
    3) Viterbi algorithm in HMMs (as in speech recognition)
    4) Value iteration for agent in stochastic environment

Exam

1. T/F conditional independence
2. Markov blankets
3. Bayes rule - word problem
4. EM algorithm in simple DAG
   - computing log-likelihood
   - posterior prob.
   - update rules for CPTs
5. Interesting BNs:
   - noisy-OR
   - naive Bayes
   - Markov and hidden Markov models
6. HMM - recursions (forward/backward), Viterbi
7. MDPs: HNG, Bellman equation, convergence, value iteration