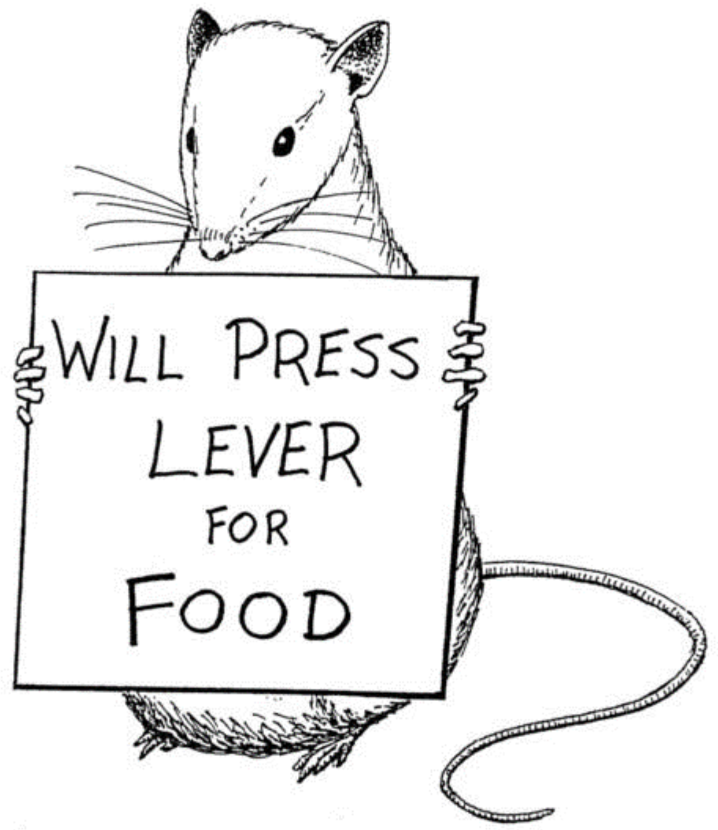
University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2021

Lecture 12: Partial-Observability Methods

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine



Today's lecture

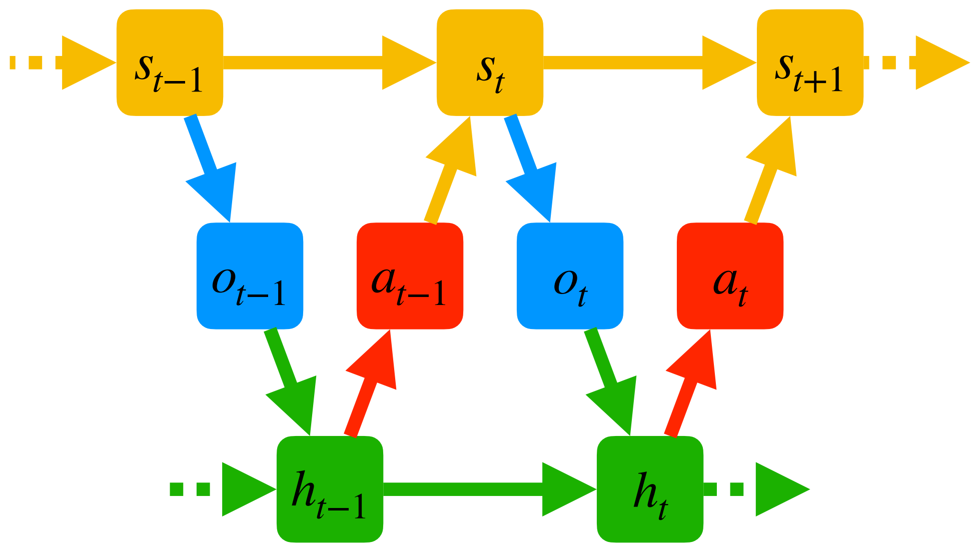
RNNs

Belief-state value function

Point-Based Value Iteration

Filtering with function approximation

- Instead of Bayesian belief, compute memory update $h_t = f_\theta(h_{t-1}, o_t)$
  - Action policy: $\pi_\theta(a_t \mid h_t)$
  - Sequential structure = Recurrent Neural Network (RNN)



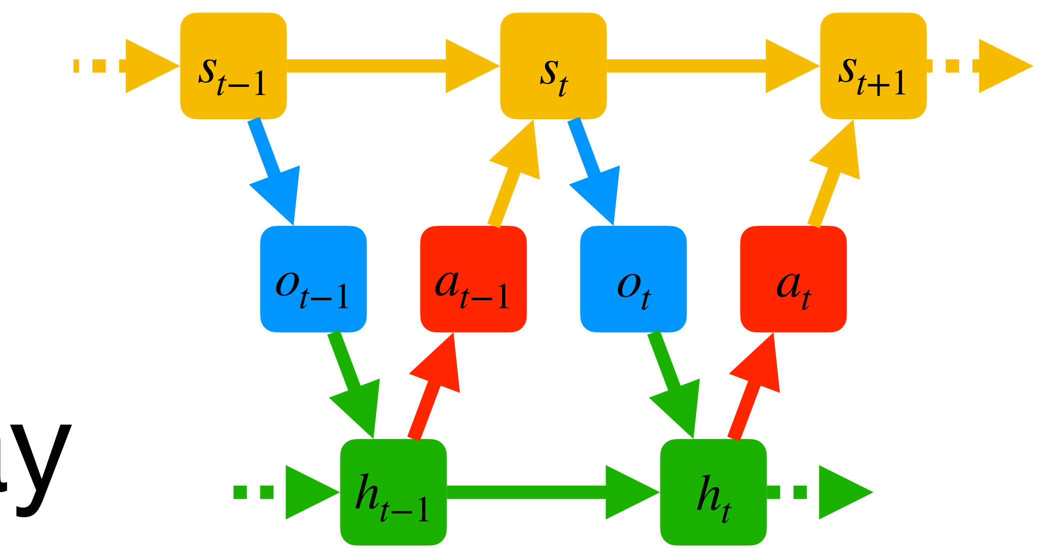
- Training = back-propagate gradients through the whole sequence
  - Back-propagation through time (BPTT)
- Unfortunately, gradients tend to vanish → 0 / explode → ∞
  - Long term coordination of memory updates + actions is challenging
  - RNN can't use information not remembered, but no memory gradient unless used

RNNs in on-policy methods

- Training RNNs with on-policy methods is straightforward (and backward)
  - Roll out policy: parameters of a_t distribution are determined by $\pi_\theta(h_t)$ with

$$h_t = f_\theta(\cdots f_\theta(f_\theta(o_0), o_1), \cdots o_t)$$

  - Compute $\nabla_\theta \log \pi_\theta(a_t | h_t)$ with BPTT all the way to initial observation o_0
- Problems: computation graph > RAM, vanishing / exploding grads
- Solution: stop gradients every k steps
- Problem: cannot learn longer memory — but that's hard anyway



RNNs in off-policy methods

- Problem: RNN states in replay buffer disagree with current RNN params

- Solution 1: use n-step rollouts

$$Q_\theta(s_t, h_t, a_t) \to r_t + \gamma r_{t+1} + \cdots + \gamma^{n-1} r_{t+n-1} + \gamma^n \max_{a'} Q_\theta(s_{t+n}, h_{t+n}, a')$$

- Solution 2: "burn in" h_t from even earlier stored steps

- In practice: RNNs rarely used
  - Stacking k frames every step $(o_{t-k+1}, \ldots, o_t)$ may help with short-term memory

Deep RL as partial observability

- Memory-based policies fail us in Deep RL, where we need them most:
  - Deep RL is inherently partially observable
- Consider what deeper layers get as input:
  - High-level / action-driven state features are not Markov!
- Memory management is a huge open problem in Deep RL
  - Actually, in other areas of ML too: NLP, time-series analysis, video processing, ...

Recap and further considerations

- Let policies depend on observable history through memory
- Memory update: Bayesian, approximate, or learned
  - Learning to update memory is one of the biggest open problems in all of ML
- Let policy be stochastic
  - Should memory be stochastic? interesting research question...
- Let policies be non-stationary if possible, otherwise learning may be unstable
  - Time-dependent policies for finite-horizon tasks
  - Periodic policies for periodic tasks

Today's lecture

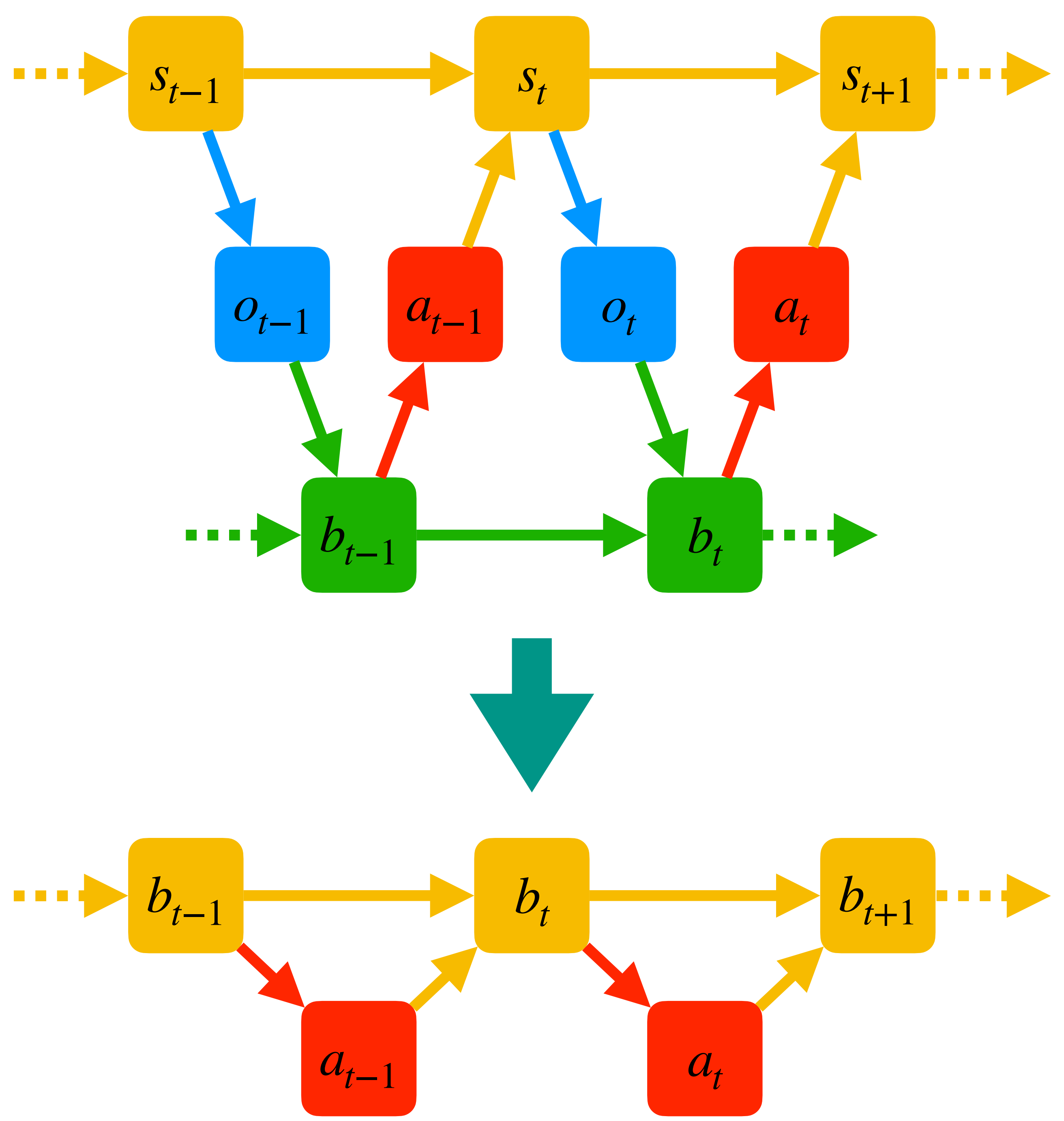
RNNs

Belief-state value function

Point-Based Value Iteration

Belief-state MDP

- Agent has seen history $h_t = (o_0, a_0, o_1, a_1, \ldots, o_t)$
  - Will see future $f_t = (a_t, o_{t+1}, \ldots, a_{T-1}, o_T)$
- State = separates past and future (given actions)
  - This means: $p(f_t | h_t, s_t) = p(f_t | s_t)$ (for fixed action seq.)



$$\implies p(f_t | h_t) = \sum_{s_t} p(s_t | h_t) p(f_t | s_t, \cancel{h_t})$$

Bayesian belief

$$= \sum_{s_t} b_t(s_t) p(f_t | s_t) = p(f_t | b_t)$$

Bayesian belief is also a state
$\implies$ all the agent needs

Belief-state value function

- If belief-states form an MDP, what is its state-value function $V_\pi(b_t) = \mathbb{E}[R_{\geq t} | b_t]$?

- Value recursion: $V_\pi(b_t) = \mathbb{E}[r(s_t, a_t) + \gamma V_\pi(b_{t+1}) | b_t]$

$$p_\pi(s_t, a_t, b_{t+1} | b_t) = b_t(s_t)\pi(a_t | b_t)p(b_{t+1} | b_t, a_t)$$

$$p(b_{t+1} | b_t, a_t) = \sum_{s_{t+1}, o_{t+1} \text{ s.t. } b_t, o_{t+1} \to b_{t+1}} p(s_{t+1} | s_t, a_t)p(o_{t+1} | s_{t+1})$$

probability of o_{t+1} leading to b_{t+1}

- $V_\pi(b_t)$ is linear in $b_t \implies V_\pi(b_t) = \sum_{s_t} b_t(s_t)\nu(s_t)$

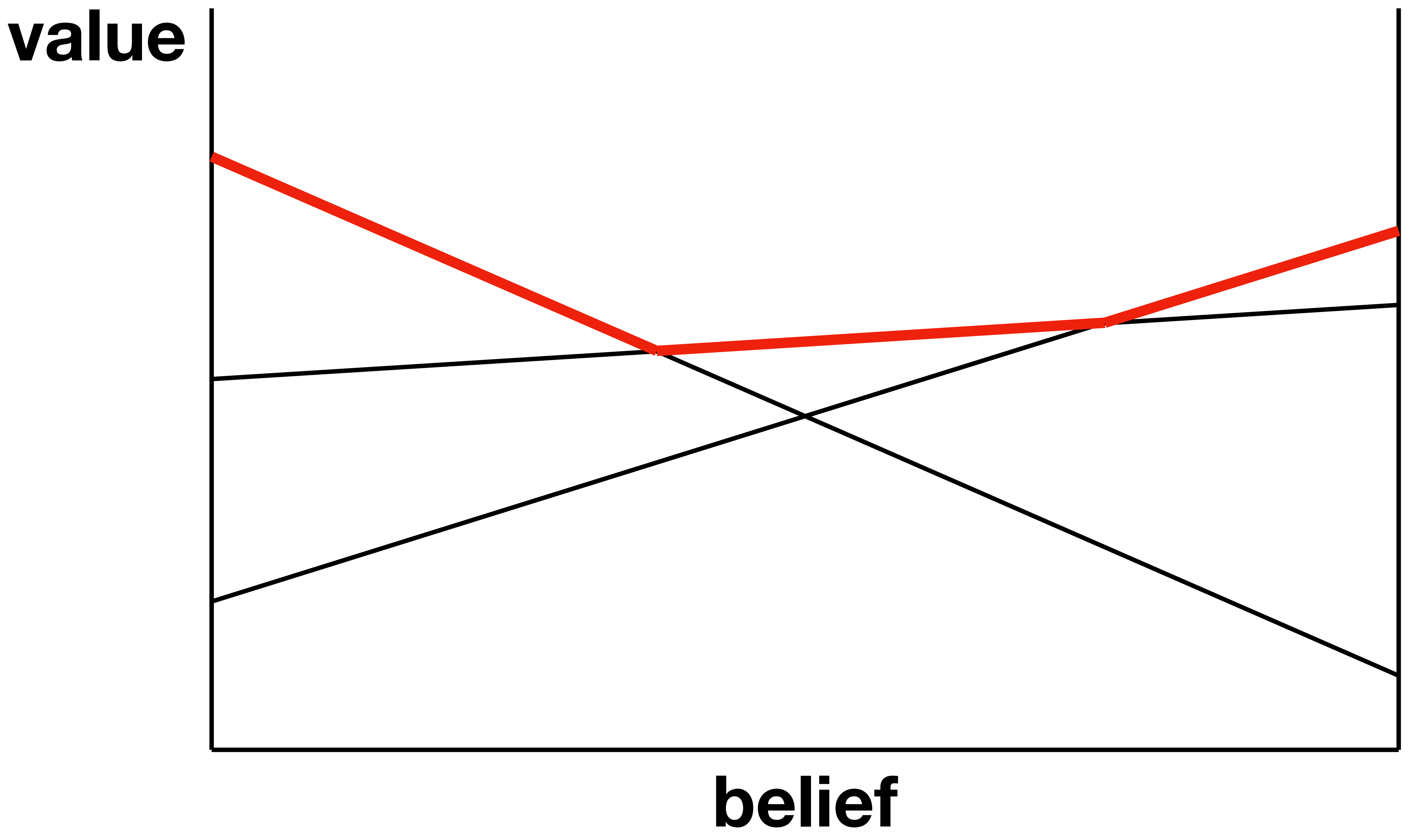
- Optimally: $V^*(b_t) = \max_{\pi \in \Pi} V_\pi(b_t) = \max_{\nu \in \mathscr{V}} \sum_{s_t} b_t(s_t) \cdot \nu(s_t) = \max_{\nu \in \mathscr{V}} b_t \cdot \nu$

linear in b_t

  - where $\mathscr{V} = \left\{\nu : \ \exists \pi \in \Pi \quad V_\pi(b_t) = b_t \cdot \nu\right\}$

Belief-state value function

- Maximum of linear functions $\implies$ piecewise-linear function



- Can be represented by set of supporting vectors $\subseteq \mathscr{V}$

First-action partitioning

- What is the structure of the belief-value support set $\mathscr{V}$?
- Let's partition by first action:

$$V^*(b_t) = \max_{a_t} Q^*(b_t, a_t)$$

linear in $p(s_t | b_t)$
$\implies$ linear in b_t

$$Q^*(b_t, a_t) = \max_{\pi} \mathbb{E}[r(s_t, a_t) + \gamma V_\pi(b_{t+1}) | b_t, a_t]$$

- $\implies \quad Q^*(b_t, a_t) = \max_{\nu \in \mathscr{V}_{a_t}} b_t \cdot \nu$
- $\implies$ We can partition $\mathscr{V}$ by first action $\mathscr{V} = \bigcup_a \mathscr{V}_a$

Next-step partition

- Recall: $b_{t+1}(s_{t+1}; b_t, a_t, o_{t+1}) = \dfrac{p(s_{t+1}, o_{t+1} \mid b_t, a_t)}{p(o_{t+1} \mid b_t, a_t)}$

Bayes' rule

$$\implies Q^*(b_t, a_t) = \max_\pi \mathbb{E}[r(s_t, a_t) + \gamma V_\pi(b_{t+1}) \mid b_t, a_t]$$

$$= \mathbb{E}[r(s_t, a_t)] + \gamma \sum_{o_{t+1}} p(o_{t+1} \mid b_t, a_t) \max_\pi V_\pi(b_{t+1})$$

$$= \mathbb{E}[r(s_t, a_t)] + \gamma \sum_{o_{t+1}} \max_{\nu' \in \mathscr{V}} \sum_{s_{t+1}} p(s_{t+1}, o_{t+1} \mid b_t, a_t) \nu'(s_{t+1})$$

$$= b_t \cdot r(\cdot\,, a_t) + \gamma \sum_{o_{t+1}} \max_{\nu \in \mathscr{V}_{a_t, o_{t+1}}} b_t \cdot \nu$$

$$\implies \mathscr{V}_a = r(\cdot\,, a) + \gamma \bigoplus_{o'} \mathscr{V}_{a,o'}$$

$\max_\pi V_\pi(b_{t+1}) = \max_{\nu' \in \mathscr{V}} b_{t+1} \cdot \nu'$

linear in $p(s_t \mid b_t)$ $\implies$ linear in b_t

sum of max = max of all combinations of sums

Value Iteration in belief-state MDP

- Represent $V(b_t)$ as $\max_{\nu \in \mathcal{V}} b_t \cdot \nu$
- Backward recursion:

$$\mathcal{V}_{a,o'} = \left\{ \nu(s) = \sum_{s'} p(s', o' | s, a) \nu'(s') : \ \nu' \in \mathcal{V} \right\}$$

$$\mathcal{V}_a = r(\cdot, a) + \gamma \bigoplus_{o'} \mathcal{V}_{a,o'}$$

$$\mathcal{V} = \bigcup_a \mathcal{V}_a$$

Today's lecture

RNNs

Belief-state value function

Point-Based Value Iteration

Representing belief value by its support

- Another curse of history: the support of $\mathscr{V}$ has at worst $|\mathscr{A}|^{|\mathscr{O}|^{T-t}}$ vectors
  - For infinite horizon, value function may even be uncomputable!
- Do we need all these ν?
  - Some may be optimal only in unreachable beliefs
  - Some may be optimal for beliefs not reached by an optimal policy
  - Some may be optimal for beliefs with low probability of being reached
  - Some may only be slightly better than others on likely beliefs

Point-Based Value Iteration (PBVI)

- Only try to optimize the value for a finite set of belief points $\mathscr{B}$
  - That means having a small subset $\mathscr{V}^{\mathscr{B}}$ of all support vectors
- We compute $\mathscr{V}^{\mathscr{B}}_{a,o'}$ from $\mathscr{V}^{\mathscr{B}}$ as before
- But now we optimize the policy suffix for a specific belief point

$$\mathscr{V}^b_a = r(\cdot, a) + \gamma \sum_{o'} \arg \max_{\nu' \in \mathscr{V}^{\mathscr{B}}_{a,o'}} b \cdot \nu'$$

- Then optimize the first action, and repeat for all belief points

$$\mathscr{V}^{\mathscr{B}} = \left\{ \arg \max_{\{\nu^b_a\}} b \cdot \nu^b_a \right\}$$

PBVI belief set expansion

- With fixed $\mathscr{B}$, repeat the approximate VI backward until near-convergence
  - This leads to approximate optimality, if $\mathscr{B}$ covers beliefs we care about
- One way to expand $\mathscr{B}$ to improve belief-space coverage:
  - For each $b \in \mathscr{B}$ and a, sample the following observation o', compute $b'(s'; b, a, s)$
  - For each $b \in \mathscr{B}$, add farthest belief from $\mathscr{B}$, in L_1 distance
- To use the solution: $\pi(b) = \arg\max_a b \cdot \nu_a^b$
- Proposition: let $\epsilon = \max_{b \text{ reachable}} \min_{b' \in \mathscr{B}} \|b' - b\|_1$ be the density of $\mathscr{B}$, then

$$\|V^* - V^{\mathscr{B}}\|_\infty \leq \frac{1}{(1-\gamma)^2} R_{\max}\epsilon$$

Learning with partial observation

- Learning with partial observation is particularly challenging
  - If we never see states, how do we know:
    - How to represent them?
    - How many there are?
  - New challenge of exploration
  - New challenge of model-selection
    - How to choose robust representations among equivalent ones?
    - How to discover the causal structure?

Learning: exponentially harder than planning

- In MDPs, we had polynomial model-based learning (E^3, R-max)
- In POMDPs, learning can be exponentially harder than planning
- Password game: guess n bits, unobservable, reward on success
  - Planning: with the dynamics known, password is known
  - Learning: have to brute-force, exponentially many guesses
- What if we can pay to observe state?
  - Too expensive for optimal policy $\Longrightarrow$ only used in training
  - Polynomial sample complexity possible in some classes

Recap

- Belief-state value function is piecewise linear
  - Can be represented by supporting vectors
  - But there are exponentially many
  - We can approximate by using a subset of the supporting vectors
    - PBVI: choose vectors by recursive optimality for beliefs we care about