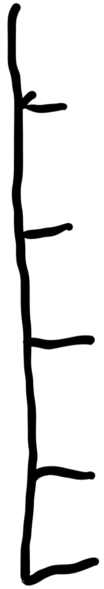
Recap

Recap

DMU



Recap

DMU

- Probabilistic Models
-
-
-
-

Recap

DMU
- Probabilistic Models
- MDPs
- Reinforcement Learning
-
-

Recap

DMU
- Probabilistic Models
- MDPs
- Reinforcement Learning
- POMDPs
- Games

Probabilistic Models

Probabilistic Models

3 Rules

$P(A)$

$P(A,B)$

$P(A|B)$

Probabilistic Models

$P(A)$

$P(A, B)$

$P(A|B)$

3 Rules

1. $0 \leq P(X \mid Y) \leq 1$

   $\sum_{x \in X} P(x \mid Y) = 1$

Probabilistic Models

$P(A)$
$P(A, B)$
$P(A \mid B)$

3 Rules

1. $0 \leq P(X \mid Y) \leq 1$
   $\sum_{x \in X} P(x \mid Y) = 1$

2. $P(X) = \sum_{y \in Y} P(X, y)$

Probabilistic Models

$P(A)$
$P(A, B)$
$P(A | B)$

3 Rules

1. $0 \leq P(X \mid Y) \leq 1$
   $\sum_{x \in X} P(x \mid Y) = 1$

2. $P(X) = \sum_{y \in Y} P(X, y)$

3. $P(X \mid Y) = \frac{P(X,Y)}{P(Y)}$

Probabilistic Models

$P(A)$
$P(A,B)$
$P(A|B)$

3 Rules

1. $0 \leq P(X \mid Y) \leq 1$
   $\sum_{x \in X} P(x \mid Y) = 1$

2. $P(X) = \sum_{y \in Y} P(X, y)$

3. $P(X \mid Y) \ = \frac{P(X,Y)}{P(Y)}$

Bayes Rule

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Probabilistic Models

$P(A)$
$P(A, B)$
$P(A | B)$

3 Rules

1. $0 \leq P(X \mid Y) \leq 1$
   $\sum_{x \in X} P(x \mid Y) = 1$

2. $P(X) = \sum_{y \in Y} P(X, y)$

3. $P(X \mid Y) = \frac{P(X,Y)}{P(Y)}$

Bayes Rule

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

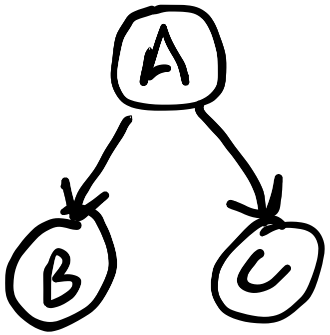
Independence

$$A \perp B \iff P(A, B) = P(A)P(B)$$

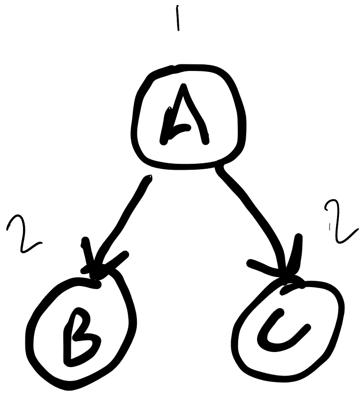
$$A \perp B \mid C \iff P(A, B \mid C) = P(A \mid C)P(B \mid C)$$

Bayesian Networks

Bayesian Networks



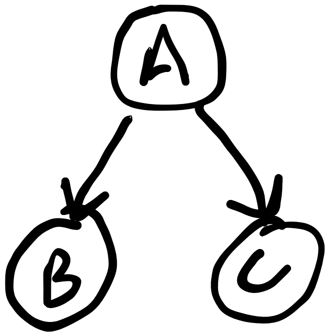
Bayesian Networks



Chain Rule

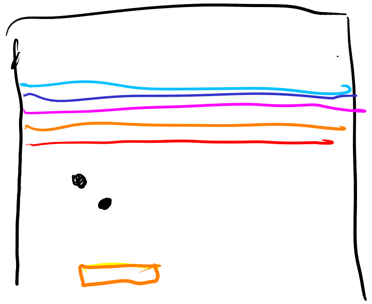
$$P(X_{1:n}) = \prod_i P(X_i \mid Pa(X_i))$$

Bayesian Networks



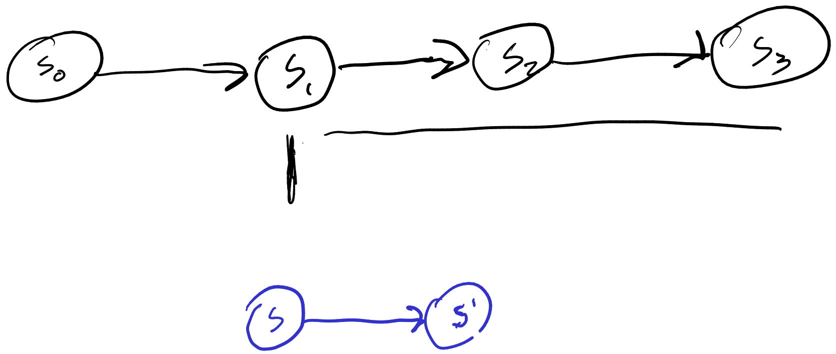
Chain Rule

$$P(X_{1:n}) = \prod_i P(X_i \mid Pa(X_i))$$

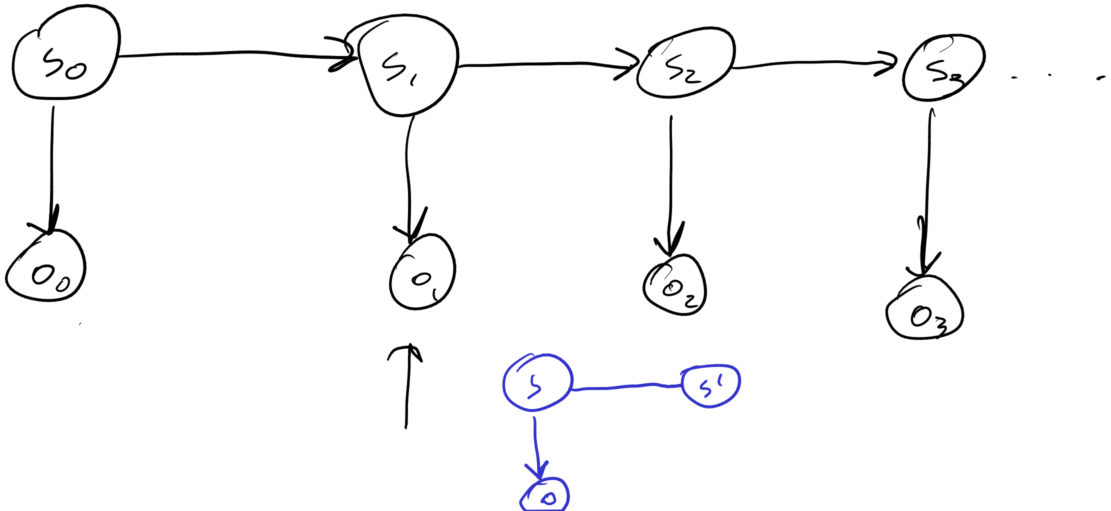


Conditional Independence

$X \perp Y \mid \mathcal{C}$ if all paths between X and Y are d-separated by $\mathcal{C}$



$o_2 \not\perp o_0 \mid o_1$



Markov Decision Processes

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Examples: $S = \{1, 2, 3\}$ or $S = \mathbb{R}^2$

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Examples: $S = \{1, 2, 3\}$ or $S = \mathbb{R}^2$

$$s = (x, \dot{x}) \in S = \mathbb{R}^2$$

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Examples: $S = \{1, 2, 3\}$ or $S = \mathbb{R}^2$

$$s = (x, \dot{x}) \in S = \mathbb{R}^2$$

$$\underset{\pi}{\text{maximize}} \; E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Examples: $S = \{1, 2, 3\}$ or $S = \mathbb{R}^2$

$$s = (x, \dot{x}) \in S = \mathbb{R}^2$$

$$\underset{\pi}{\text{maximize}} \; E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

$$Q^\pi(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a, a_t = \pi(s_t)\right]$$

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Examples: $S = \{1, 2, 3\}$ or $S = \mathbb{R}^2$

$$s = (x, \dot{x}) \in S = \mathbb{R}^2$$

$$\underset{\pi}{\text{maximize}} \; E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

$$Q^\pi(s, a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a, a_t = \pi(s_t)\right]$$

$$V^\pi(s) = Q^\pi(s, \pi(s))$$

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Examples: $S = \{1, 2, 3\}$ or $S = \mathbb{R}^2$

$$s = (x, \dot{x}) \in S = \mathbb{R}^2$$

$$\underset{\pi}{\text{maximize}} \; E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

$$Q^\pi(s, a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a, a_t = \pi(s_t)\right]$$

$$V^\pi(s) = Q^\pi(s, \pi(s))$$

$$V^\pi(s) = R(s, \overset{\pi(s)}{\cancel{a}}) + \gamma E[V^\pi(s')]$$ ← policy evaluation

$$V^*(s) = \max_a \{R(s, a) + \gamma E[V^*(s')]\}$$ ← certificate of optimality "Bellman's Eq."

$$B[V](s) = \max_a \{R(s, a) + \gamma E[V(s')]\}$$ ← Bellman's operator

Markov Decision Processes

$$(S, A, R, T, \gamma)$$

Examples: $S = \{1, 2, 3\}$ or $S = \mathbb{R}^2$

$$s = (x, \dot{x}) \in S = \mathbb{R}^2$$

$$\text{maximize}_{\pi} \; E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

$$Q^\pi(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a, a_t = \pi(s_t)\right]$$

$$V^\pi(s) = Q^\pi(s, \pi(s))$$

$$V^\pi(s) = R(s,a) + \gamma E[V^\pi(s')]$$ Policy Evaluation

$$V^*(s) = \max_a \{R(s,a) + \gamma E[V^*(s')]\}$$ Bellman's Equation: Certificate of Optimality

$$B[V](s) = \max_a \{R(s,a) + \gamma E[V(s')]\}$$ Bellman's Operator

Offline MDP Algorithms

Offline MDP Algorithms

Policy Iteration

loop

    Evaluate Policy

    Improve Policy

Offline MDP Algorithms

Policy Iteration

loop

  Evaluate Policy

  Improve Policy

Value Iteration

loop

  $V \leftarrow B[V]$

Offline MDP Algorithms

Policy Iteration

loop

    Evaluate Policy

    Improve Policy

Converges because policy always improves and there are a finite number of policies

Value Iteration

loop

    $V \leftarrow B[V]$

Offline MDP Algorithms

Policy Iteration

loop

Evaluate Policy

Improve Policy

Converges because policy always improves and there are a finite number of policies

Value Iteration

loop

$V \leftarrow B[V]$

Converges because B is a contraction mapping

Curse of Dimensionality $\rightarrow$ Large $|S|$

Online MDP Planning

Online MDP Planning

Monte Carlo Tree Search

Online MDP Planning

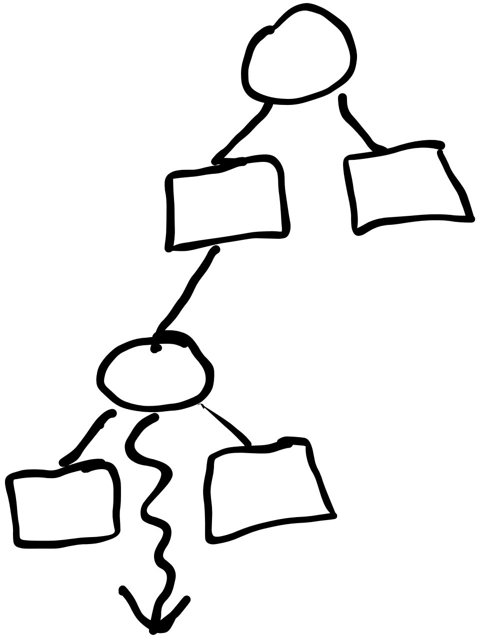
Monte Carlo Tree Search

Search

Expand

Rollout

Backup



Online MDP Planning

Monte Carlo Tree Search

Search

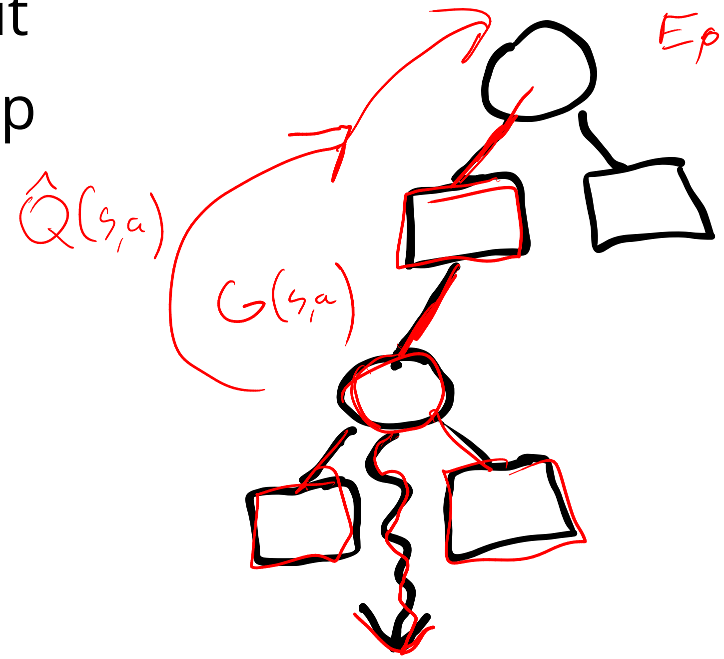
Expand

Rollout

Backup

Valuable

$$Q(s,a) + \underbrace{c\sqrt{\frac{\log N(s)}{N(s,a)}}}_{\text{Exploration Bonus}}$$



maximize $\hat{Q}(s,a)$ at root

or

maximize $N(s,a)$

Online MDP Planning

Monte Carlo Tree Search

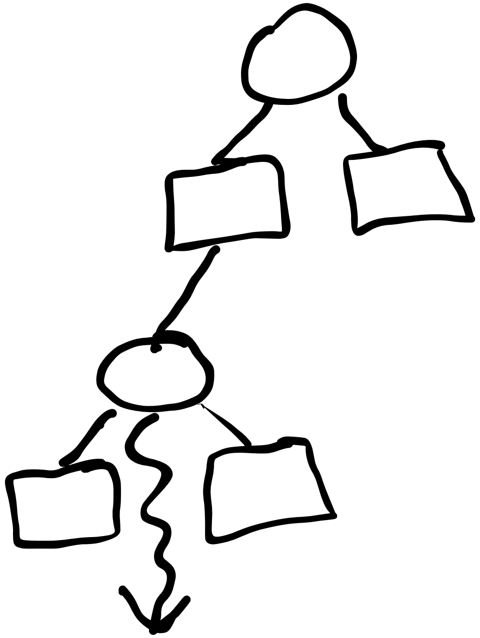
Search

Expand

Rollout

Backup

$$Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}}$$



Sparse Sampling

Online MDP Planning

Monte Carlo Tree Search

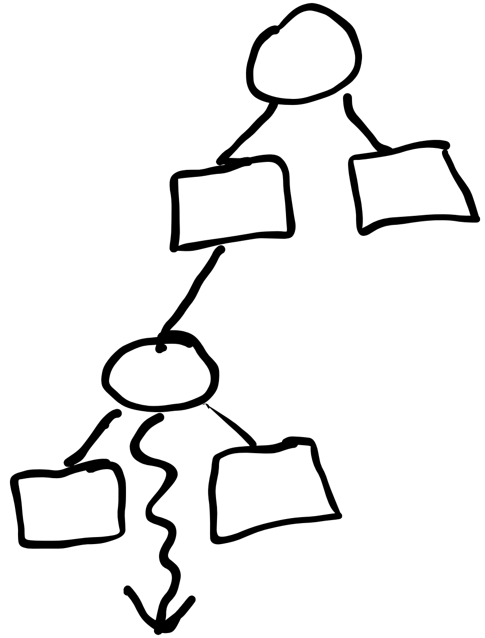
Search

Expand

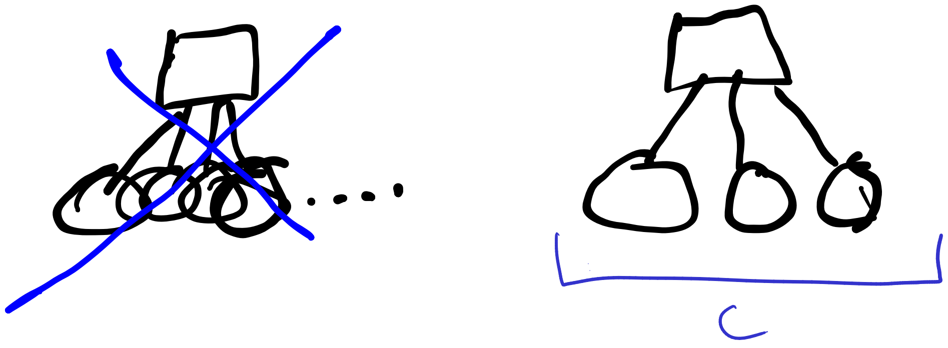
Rollout

Backup

$$Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}}$$



Sparse Sampling



Online MDP Planning

Monte Carlo Tree Search

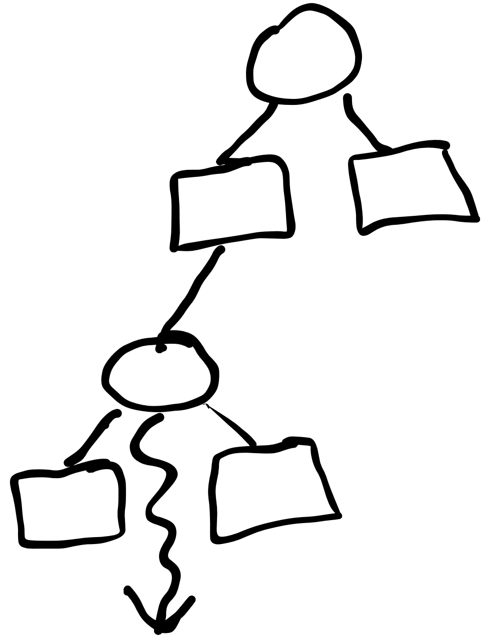
Search

Expand

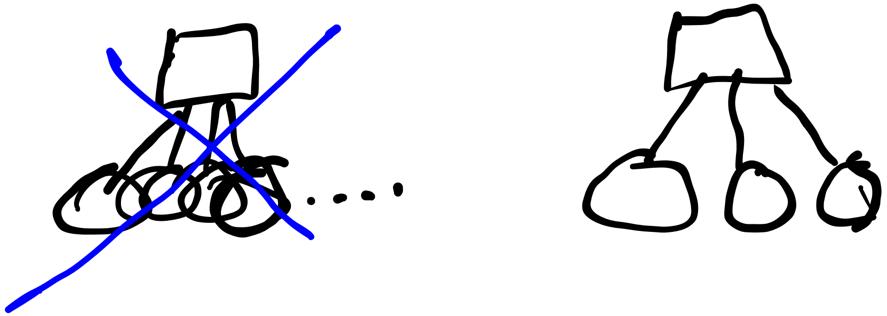
Rollout

Backup

$$Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}}$$



Sparse Sampling

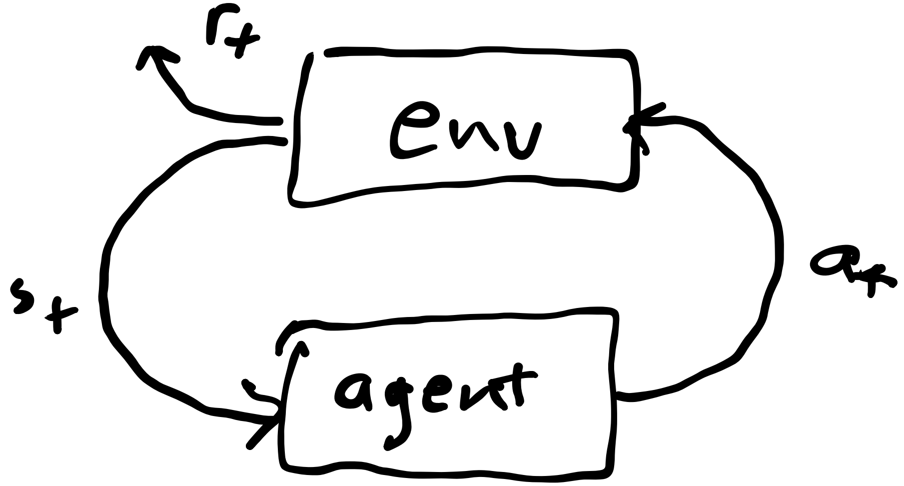


Guarantees *independent* of $|S|$!!

Reinforcement Learning

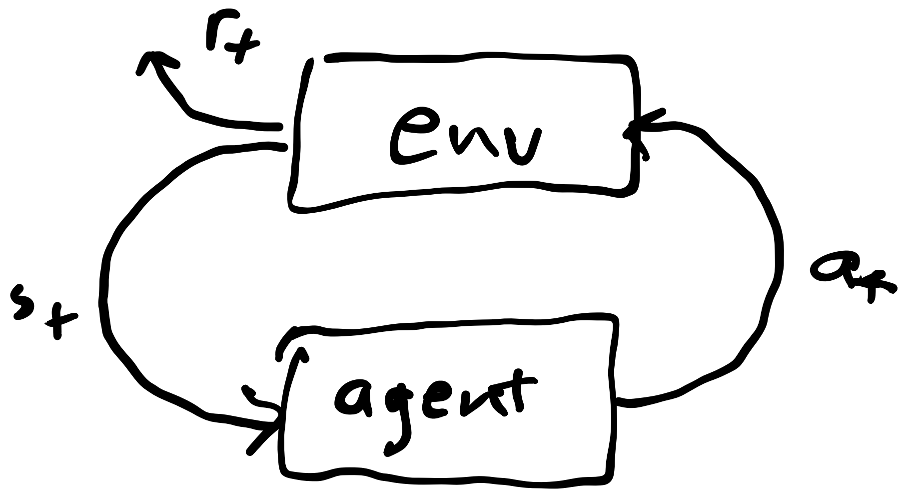
Challenges:

Reinforcement Learning



Challenges:

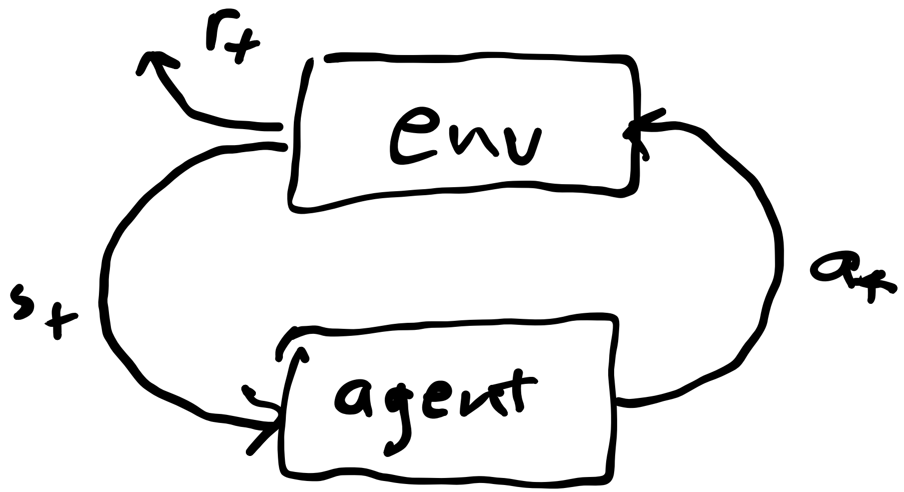
Reinforcement Learning



Challenges:

1. Exploration and Exploitation

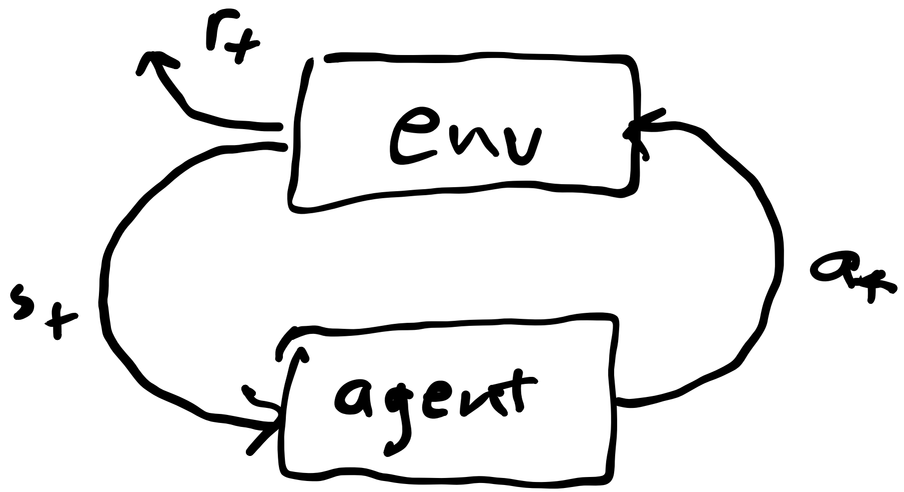
Reinforcement Learning



Challenges:

1. Exploration and Exploitation
2. Credit Assignment

Reinforcement Learning



Challenges:

1. Exploration and Exploitation
2. Credit Assignment
3. Generalization

Exploration

Exploration

Bandits

easiest

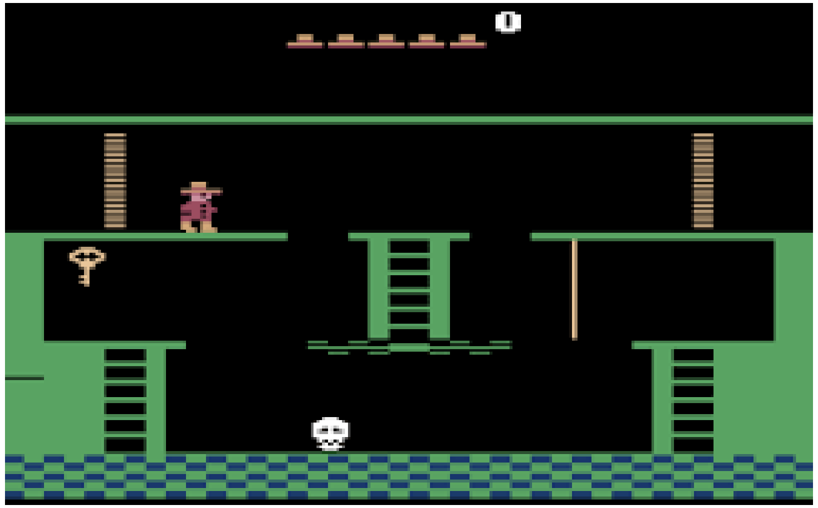
- ϵ-greedy
- softmax
- UCB $\hat{Q}(a) + c\sqrt{\frac{\log N}{N(a)}}$
- Thompson Sampling
- Optimal DP Solution (solving a POMDP!)

logarithmic regret

Exploration

Bandits

- ϵ-greedy
- softmax
- UCB
- Thompson Sampling
- Optimal DP Solution (solving a POMDP!)

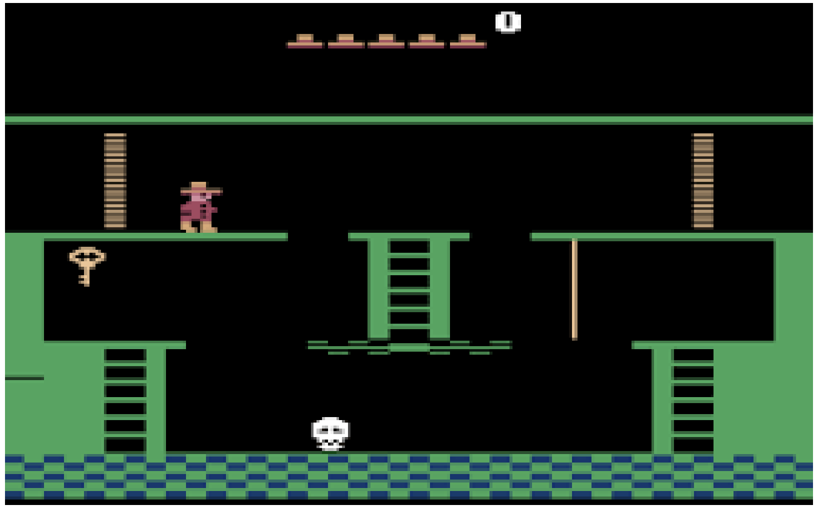


Montezuma's Revenge!

Exploration

Bandits

- ϵ-greedy
- softmax
- UCB
- Thompson Sampling
- Optimal DP Solution (solving a POMDP!)



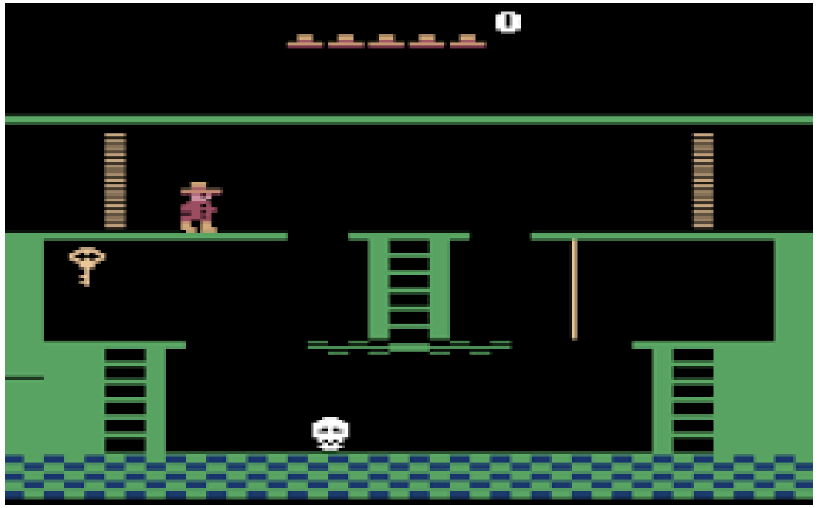
Montezuma's Revenge!

- Pseudocounts

Exploration

Bandits

- ϵ-greedy
- softmax
- UCB
- Thompson Sampling
- Optimal DP Solution (solving a POMDP!)



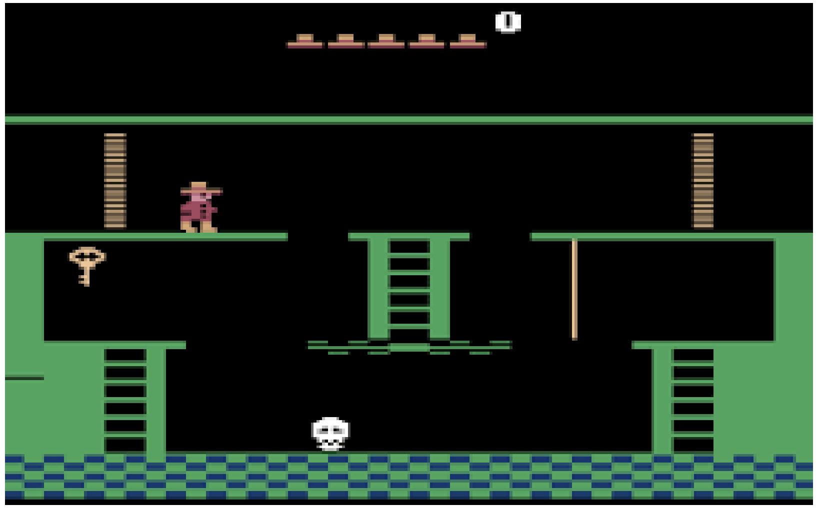
Montezuma's Revenge!

- Pseudocounts
- Curiosity: extra reward for bad prediction

Exploration

Bandits

- ϵ-greedy
- softmax
- UCB $Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}}$
- Thompson Sampling
- Optimal DP Solution (solving a POMDP!)



Montezuma's Revenge!

- Pseudocounts
- Curiosity: extra reward for bad prediction
- Random network distillation

$s' = f_\theta(s,a)$

$f_\phi(s,a) = f_\theta(s,a)$

RL Algorithms

RL Algorithms

Model Based

Model Free

RL Algorithms

Model Based

Model Free

On Policy

Off Policy

RL Algorithms

Model Based

Model Free

On Policy

Off Policy

MLMBTRL
(learn T,R)

RL Algorithms

Model Based

Model Free

learn Q

learn π

On Policy

Off Policy

MLMBTRL
(learn T,R)

RL Algorithms

Model Based

Model Free

learn Q

learn π

Policy Gradient

On Policy

Off Policy

MLMBTRL
(learn T,R)

RL Algorithms

Model Based

Model Free

learn Q

SARSA

learn π

Policy Gradient

On Policy

Off Policy

MLMBTRL
(learn T, R)

RL Algorithms

Model Based ⟷ Model Free

learn Q	learn π
SARSA	Policy Gradient

On Policy ↕ Off Policy

MLMBTRL (learn T, R)

Q-learning

RL Algorithms

Model Based

Model Free

learn Q

SARSA

Q-learning

learn π

Policy Gradient

On Policy

Off Policy

MLMBTRL
(learn T, R)

BAMDP
(POMDP $\tilde{s} = (s, T, R)$)

Policy Gradient

Policy Gradient

- Likelihood ratio trick

$\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)$

Policy Gradient

- Likelihood ratio trick
- Causality

$\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)$

Policy Gradient

- Likelihood ratio trick
- Causality
- Baseline Subtraction

$$\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)$$

Policy Gradient

- Likelihood ratio trick
- Causality
- Baseline Subtraction

$\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)$

$$\nabla U(\theta) = \mathbb{E}_\tau \left[\sum_{k=1}^{d} \nabla_\theta \log \pi_\theta(a^{(k)} \mid s^{(k)}) \gamma^{k-1} \left(r_{\text{to-go}}^{(k)} - r_{\text{base}}(s^{(k)}) \right) \right]$$

Policy Gradient

- Likelihood ratio trick
- Causality
- Baseline Subtraction

$\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)$

$$\nabla U(\theta) = \mathbb{E}_\tau \left[\sum_{k=1}^{d} \nabla_\theta \log \pi_\theta(a^{(k)} \mid s^{(k)}) \gamma^{k-1} \left(r_{\text{to-go}}^{(k)} - r_{\text{base}}(s^{(k)}) \right) \right]$$

- Natural Gradient

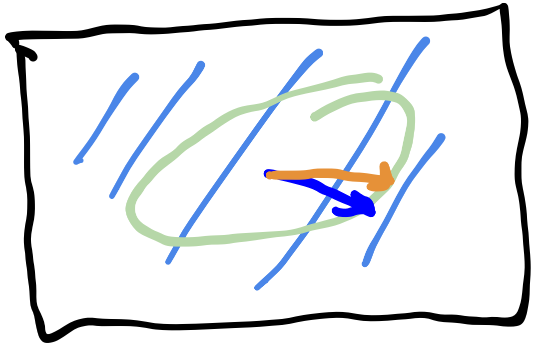
Policy Gradient

- Likelihood ratio trick
- Causality
- Baseline Subtraction

$$\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)$$

$$\nabla U(\theta) = \mathbb{E}_\tau\left[\sum_{k=1}^{d} \nabla_\theta \log \pi_\theta(a^{(k)} \mid s^{(k)}) \gamma^{k-1} \left(r_{\text{to-go}}^{(k)} - r_{\text{base}}(s^{(k)})\right)\right]$$

- Natural Gradient



KL div.
Bound

Q-Learning

Q-Learning

SARSA

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r_t + \gamma Q(s',a') - Q(s,a))$$

Q-Learning

SARSA

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r_t + \gamma Q(s',a') - Q(s,a))$$

Eligibility Traces

Q-Learning

SARSA

chosen by current policy

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r_t + \gamma \underline{Q(s',\underline{a'})} - Q(s,a))$$

Eligibility Traces

Q-learning

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r_t + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

Q-Learning

SARSA

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r_t + \gamma Q(s',a') - Q(s,a))$$

Eligibility Traces

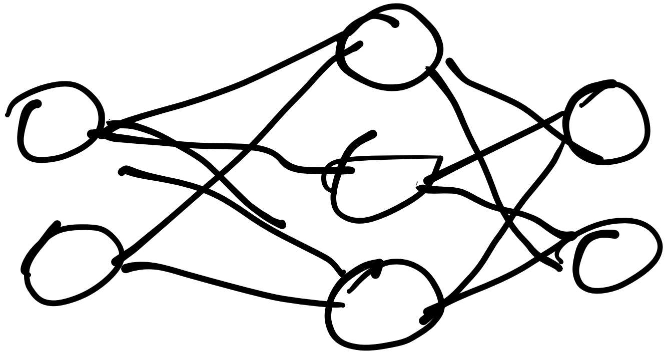
Q-learning

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r_t + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

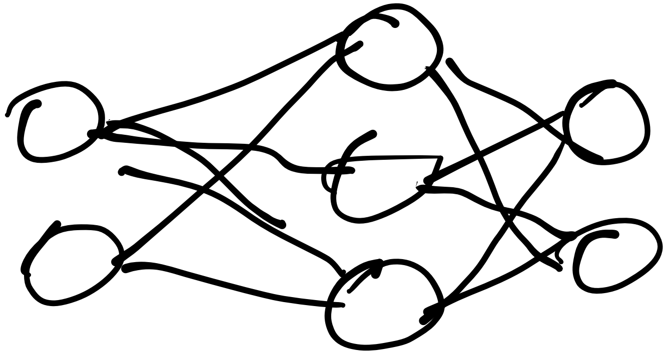
Double Q Learning

Neural Networks and DQN

Neural Networks and DQN

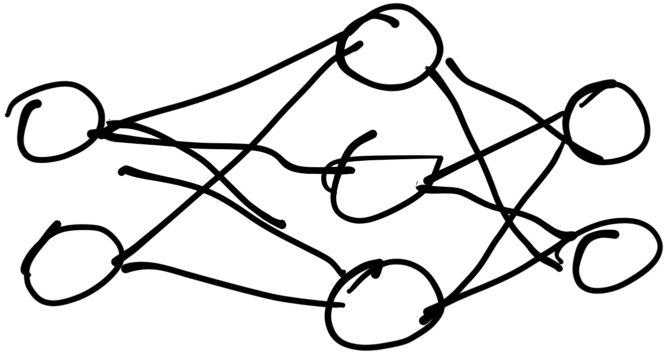


Neural Networks and DQN



$$f_\theta(x) = \sigma(W_2\sigma(W_1 x + b_1) + b_2)$$

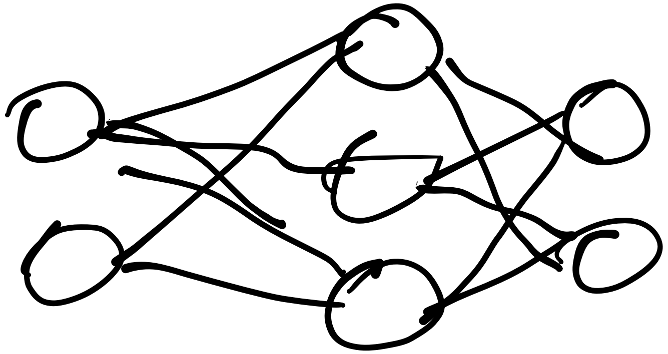
Neural Networks and DQN



$$f_\theta(x) = \sigma(W_2\sigma(W_1 x + b_1) + b_2)$$

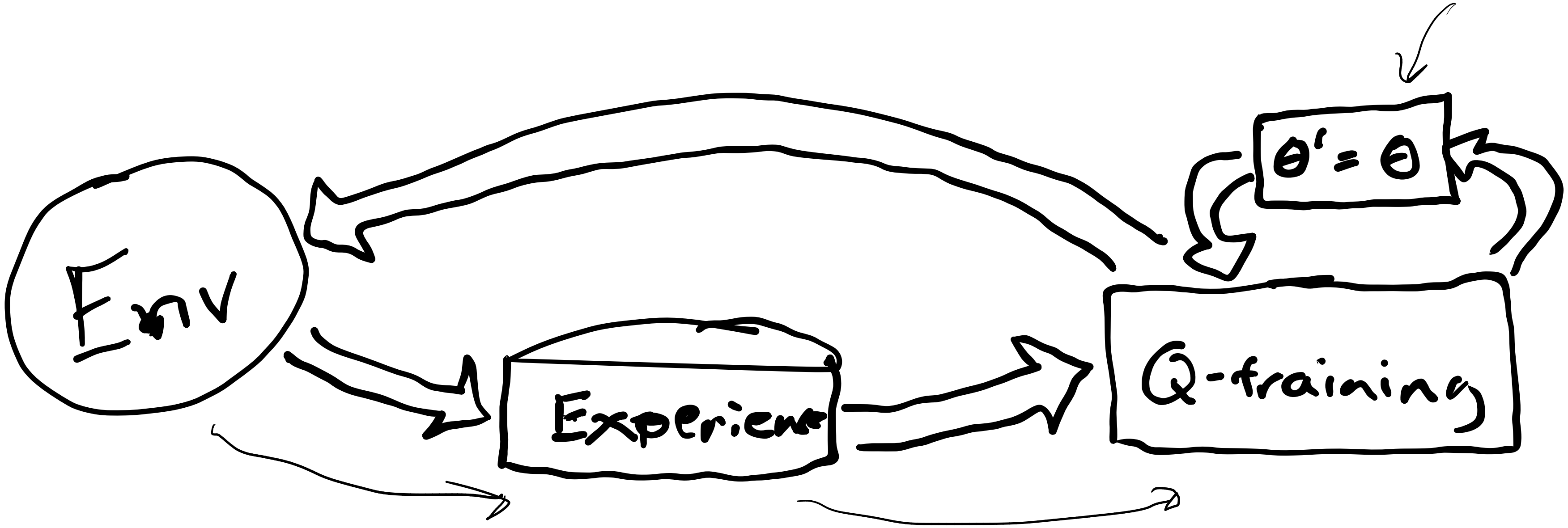
Backprop

Neural Networks and DQN



$$f_\theta(x) = \sigma(W_2\sigma(W_1x + b_1) + b_2)$$

Backprop



Actor-Critic

Actor-Critic

- Actor: π_θ

Actor-Critic

- Actor: π_θ
- Critic: Q_ϕ

Actor-Critic

- Actor: π_θ
- Critic: Q_ϕ

Soft Actor Critic

Actor-Critic

- Actor: π_θ
- Critic: Q_ϕ

Soft Actor Critic

$$J(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha \underline{\mathcal{H}(\pi(\cdot \mid s_t))}\right)\right]$$

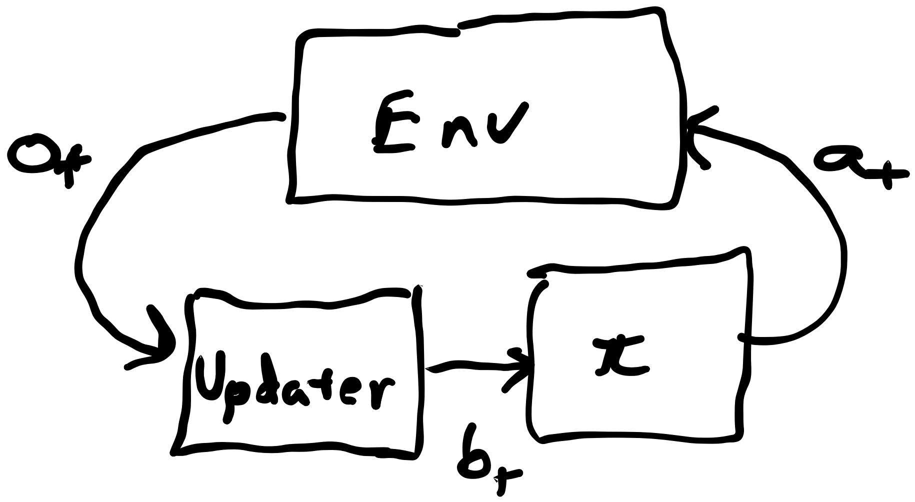
POMDPs

POMDPs

$$(S, A, T, R, O, Z, \gamma)$$

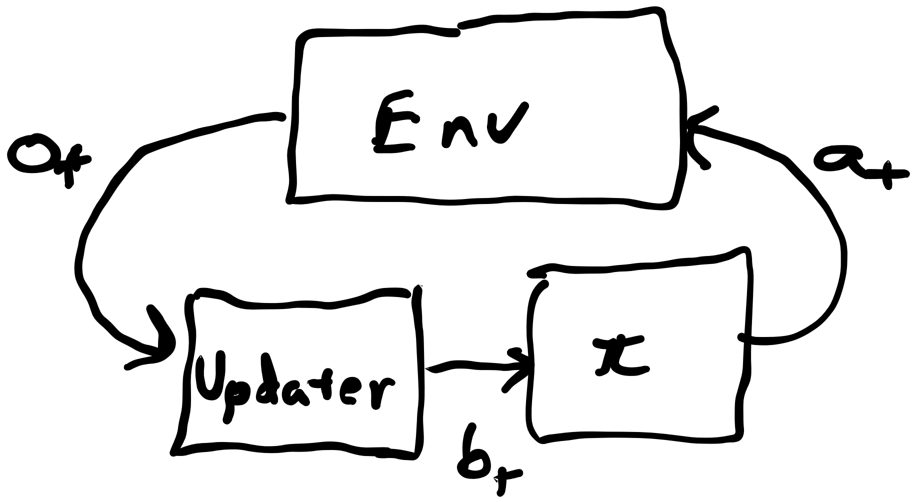
POMDPs

$$(S, A, T, R, O, Z, \gamma)$$



POMDPs

$$(S, A, T, R, O, Z, \gamma)$$

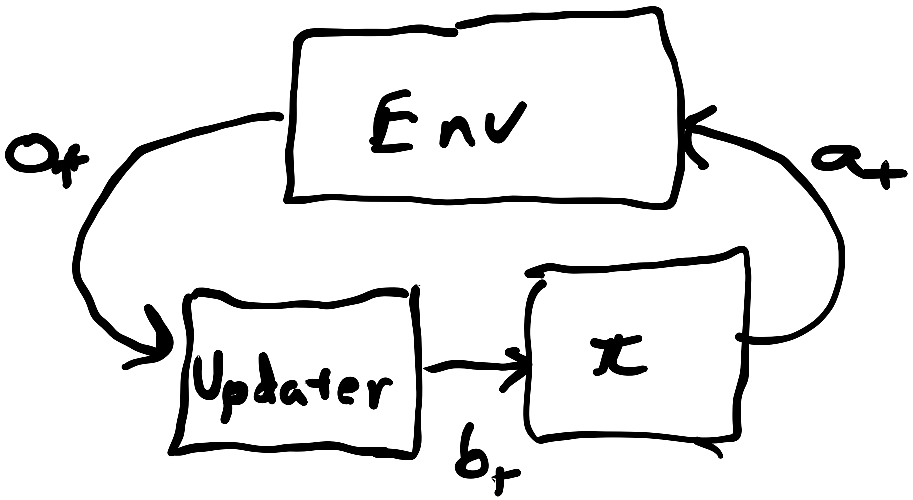


Belief Updates

- Discrete Bayesian Filter
- Particle Filter

POMDPs

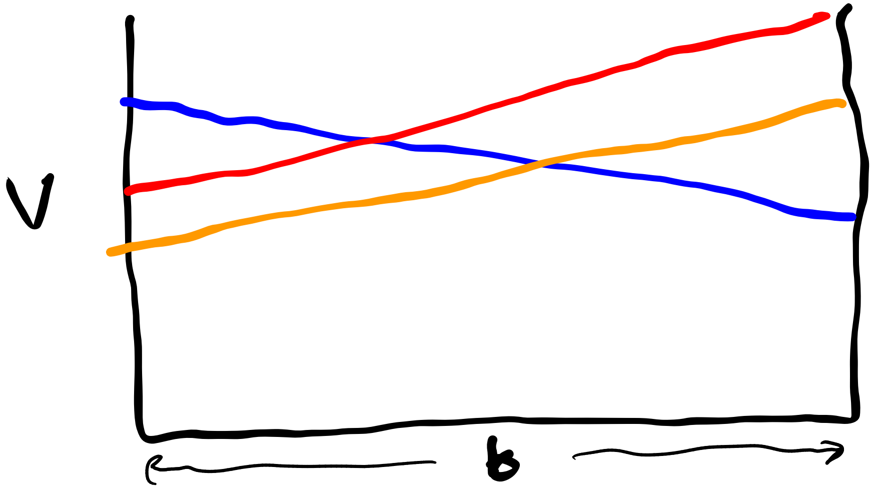
$$(S, A, T, R, O, Z, \gamma)$$



Belief Updates

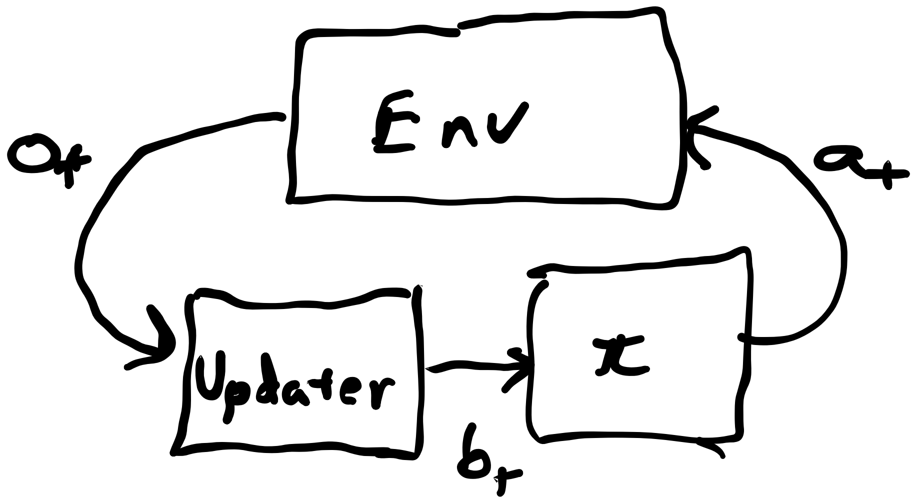
- Discrete Bayesian Filter
- Particle Filter

Alpha Vectors



POMDPs

$$(S, A, T, R, O, Z, \gamma)$$

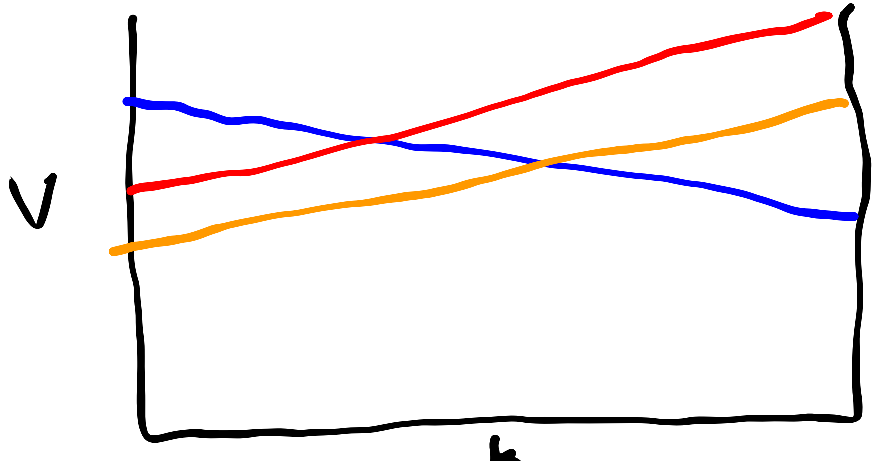


- Each alpha vector corresponds to a conditional plan

Belief Updates

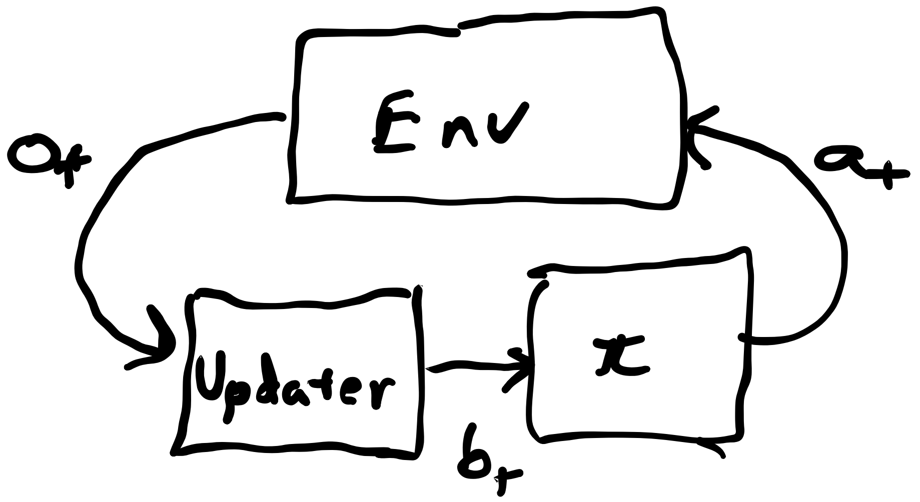
- Discrete Bayesian Filter
- Particle Filter

Alpha Vectors



POMDPs

$$(S, A, T, R, O, Z, \gamma)$$

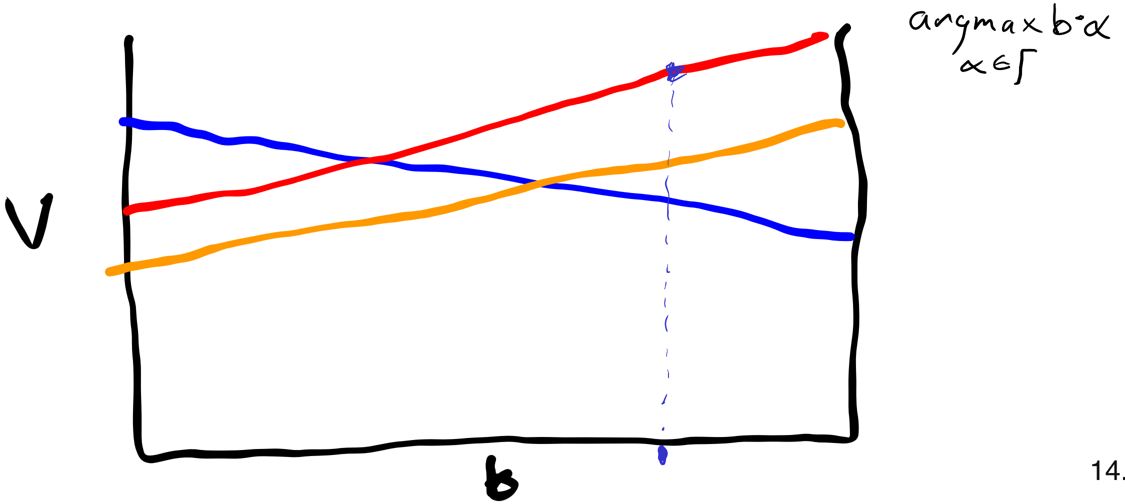


- Each alpha vector corresponds to a conditional plan
- You can prune alpha vectors by solving an LP

Belief Updates

- Discrete Bayesian Filter
- Particle Filter

Alpha Vectors



POMDP Approximations

POMDP Approximations

Formulation

- Certainty Equivalence
- QMDP

$\pi_{MDP}\left(\underset{s\sim b}{E}[s]\right)$

$\operatorname{argmax}_s b(s)$

optimal for LQG

$\pi_{QMDP}(b) = \underset{a}{\operatorname{argmax}} \underset{s\sim b}{E}[Q_{MDP}(s,a)]$

POMDP Approximations

Formulation

- Certainty Equivalence
- QMDP

Numerical

POMDP Approximations

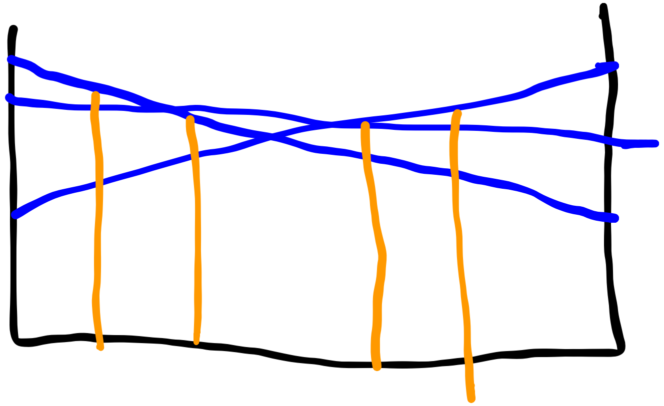
Formulation

- Certainty Equivalence
- QMDP

Numerical

Offline

- Point-Based Value Iteration
- SARSOP



POMDP Approximations

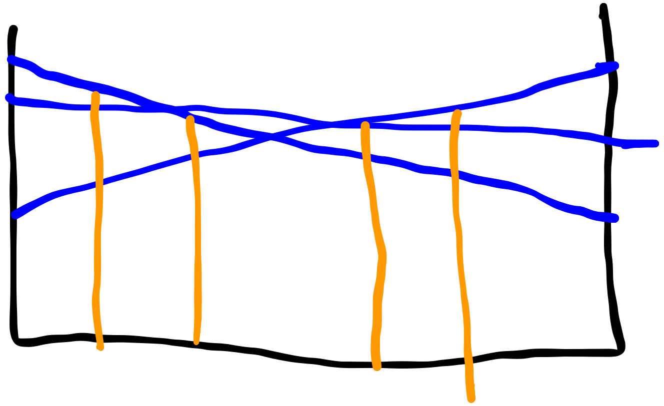
Formulation

- Certainty Equivalence
- QMDP

Numerical

Offline

- Point-Based Value Iteration
- SARSOP



Online

- POMCP
- DESPOT

POMDP Approximations

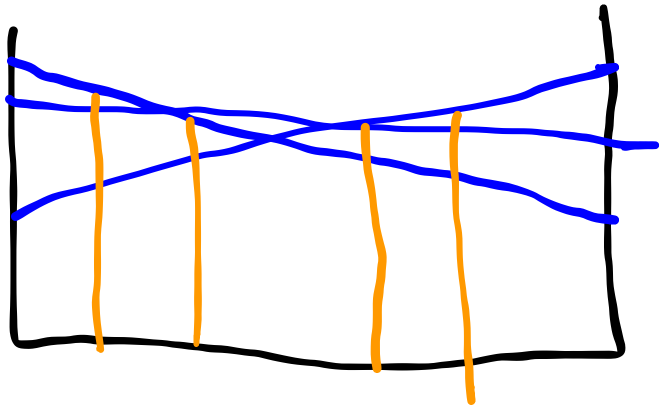
Formulation

- Certainty Equivalence
- QMDP

Numerical

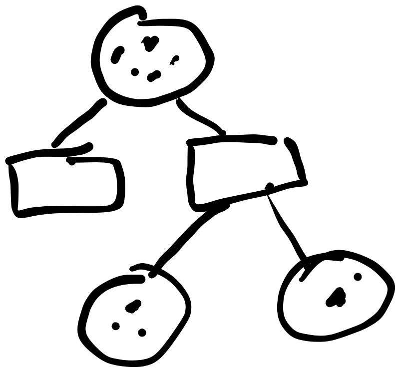
Offline

- Point-Based Value Iteration
- SARSOP



Online

- POMCP
- DESPOT



Simple Games

Simple Games

- Optimal Solutions

Simple Games

- ~~Optimal Solutions~~ No!

Simple Games

- ~~Optimal Solutions~~ No!
- Equilibria (e.g. Nash Equilibria)

Simple Games

- ~~Optimal Solutions~~ No!
- Equilibria (e.g. Nash Equilibria)

-1,-1	-3,0
0,-3	-2,-2

Simple Games

- ~~Optimal Solutions~~ No!
- Equilibria (e.g. Nash Equilibria)

-1,-1	-3,0
0,-3	-2,-2

- Every finite game has at least 1 Nash Equilibrium

Simple Games

- ~~Optimal Solutions~~ No!
- Equilibria (e.g. Nash Equilibria)

-1,-1	-3,0
0,-3	-2,-2

- Every finite game has at least 1 Nash Equilibrium
- Might be pure or mixed

Simple Games

- ~~Optimal Solutions~~ No!
- Equilibria (e.g. Nash Equilibria)

NE: every player plays a best response

	L	R
up	-1,-1	-3,0
down	0,-3	$\boxed{-2,-2}$

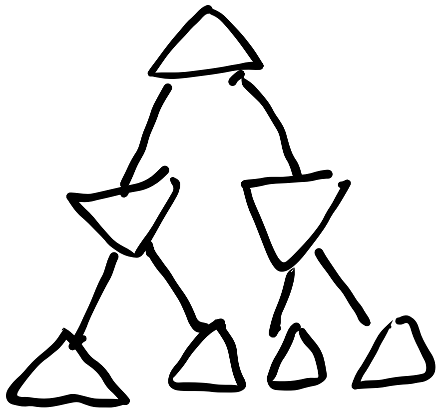
- Every finite game has at least 1 Nash Equilibrium
- Might be pure or mixed
- Algorithms like fictitious play converge in special cases

Dominant Strategy Equilibrium

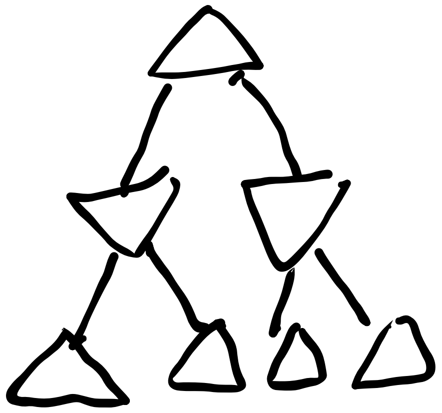
Correlated Equilibrium → NE is a correlated Eq. where all policies are independent

Turn Taking Games

Turn Taking Games

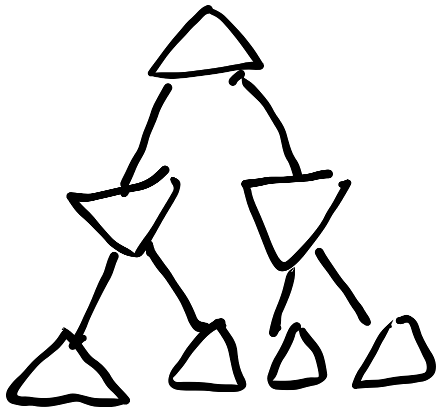


Turn Taking Games



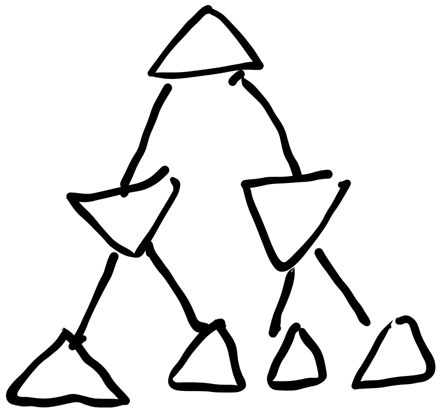
- Value Function Backup

Turn Taking Games



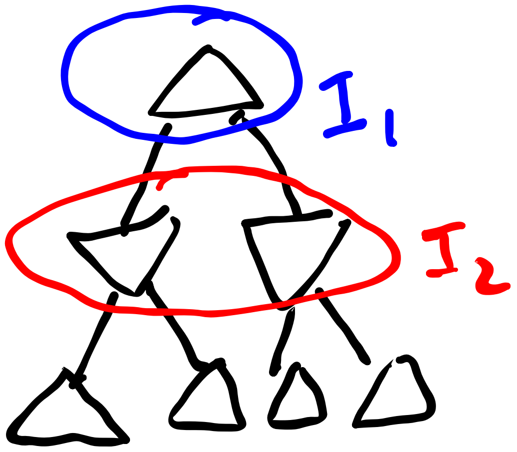
- Value Function Backup
- $\alpha\beta$ Pruning

Turn Taking Games



- Value Function Backup
- $\alpha\beta$ Pruning
- Incomplete Information Extensive Form

Turn Taking Games



- Value Function Backup
- $\alpha\beta$ Pruning
- Incomplete Information Extensive Form

Markov Games and POMGS

Markov Games and POMGS

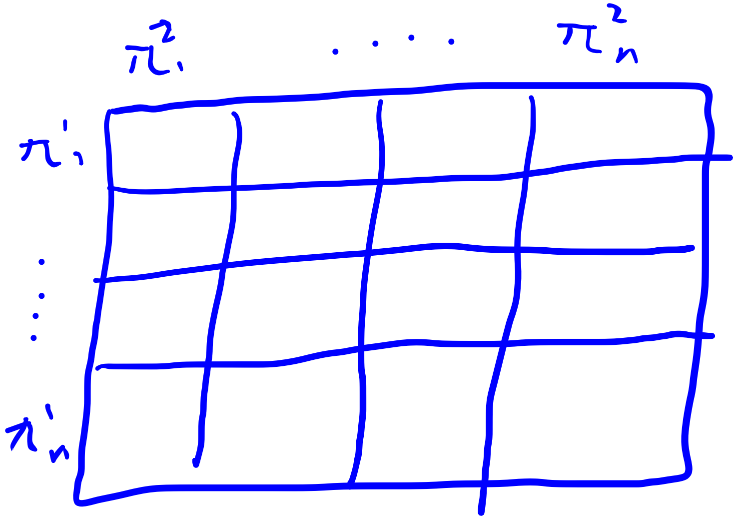
Markov Games

- All players play simultaneously
- Transitions are stochastic
- Best response involves solving an MDP
- Can be reduced to a simple game with policies as actions

Markov Games and POMGS

Markov Games

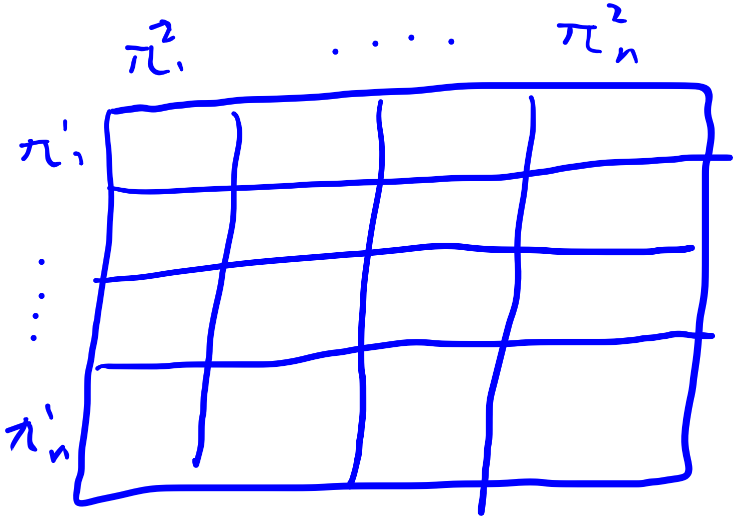
- All players play simultaneously
- Transitions are stochastic
- Best response involves solving an MDP
- Can be reduced to a simple game with policies as actions



Markov Games and POMGS

Markov Games

- All players play simultaneously
- Transitions are stochastic
- Best response involves solving an MDP
- Can be reduced to a simple game with policies as actions



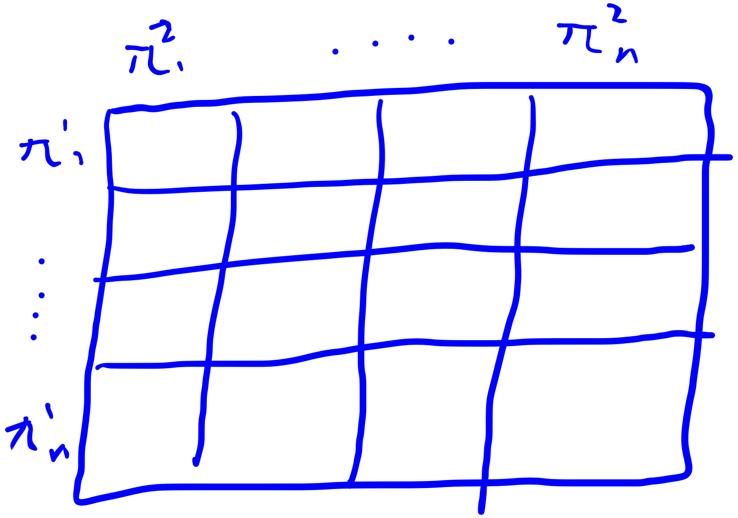
Partially Observable Markov Games

- Each player receives a noisy observation at each step
- Beliefs not practical to compute
- Can be reduced to simple game with policies as actions

Markov Games and POMGS

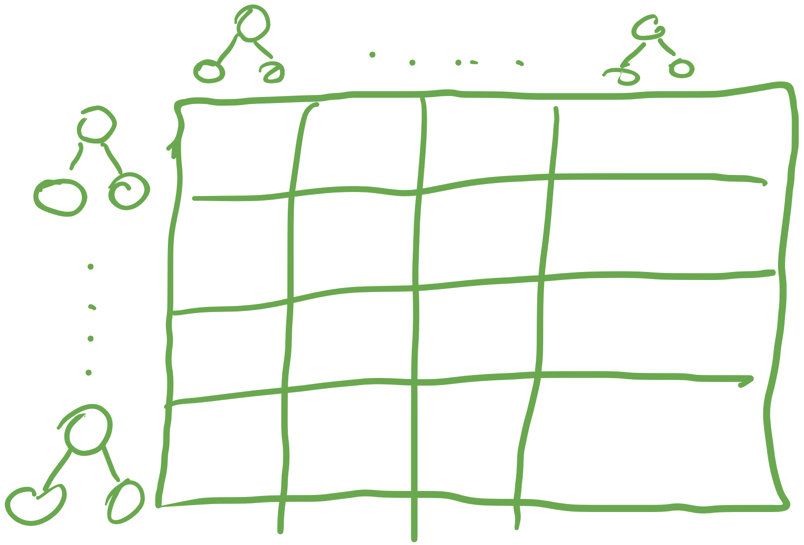
Markov Games

- All players play simultaneously
- Transitions are stochastic
- Best response involves solving an MDP
- Can be reduced to a simple game with policies as actions



Partially Observable Markov Games

- Each player receives a noisy observation at each step
- Beliefs not practical to compute
- Can be reduced to simple game with policies as actions



Fictitious Play in Markov Games

initialize $N(j, a^j, s)$ to 0
initialize π^{BR}

Loop:

1. Simulate MG with π^{BR}
2. Update $N(j, a^j, s)$ ← N should reflect all past simulations
3. $\pi^j(a^j | s) \propto N(j, a^j, s) \quad \forall j$
4. π^{BR} ← best response to π ← converges to NE for some classes of games

exploitability

solve an MDP with other player's strategies integrated into it

will π^{BR} converge? No

Recap

Recap

After DMU you have basic tools to deal with 4 Big Problems:

Recap

After DMU you have basic tools to deal with 4 Big Problems:

1. Immediate and Future Rewards

Recap

After DMU you have basic tools to deal with 4 Big Problems:

1. Immediate and Future Rewards
2. Unknown Models

Recap

After DMU you have basic tools to deal with 4 Big Problems:

1. Immediate and Future Rewards
2. Unknown Models
3. Partial Observability

Recap

After DMU you have basic tools to deal with 4 Big Problems:

1. Immediate and Future Rewards
2. Unknown Models
3. Partial Observability
4. Other Agents