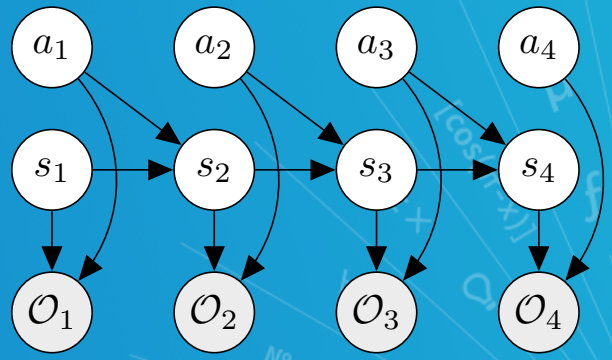
PETUUM

Carnegie Mellon University
School of Computer Science

# Probabilistic Graphical Models

## Reinforcement Learning & Control Through Inference in GM




Maruan Al-Shedivat

Lecture 20, April 1, 2019

**Reading: see class homepage**

# What we've covered so far

Module 1: Representation & Exact Inference

Module 2: Approximate Inference (VI, MC, SMC, MCMC)

Module 3 [introduced in 2015]: Deep Learning & Generative Models

Module 4 [new and experimental]: RL and Control as Inference in GM

Goal: gentle introduction to basic concepts of RL with a focus on connections between control and inference in a probabilistic GM



# A note on materials used in this module

- Sutton & Barto. **Reinforcement Learning: An Introduction**. 2nd edition.
- David Silver's UCL course on reinforcement learning.
- Materials from UC Berkeley's Deep RL course.
- Sergey Levine's tutorial on RL and control as inference.
- Brian Ziebart's PhD thesis (maximum causal entropy models).



# RL has already come up previously (in text generation)

- The generalized ERPO objective [Tan, Hu, et al., 2018]:

$$\mathcal{L}(q,\theta) = \mathbb{E}_q[R(\boldsymbol{y} \mid \boldsymbol{y}^*)] - \alpha\,\mathrm{KL}[q(\boldsymbol{y} \mid \boldsymbol{x}) || p_\theta(\boldsymbol{y} \mid \boldsymbol{x})] + \beta\,\mathrm{H}(q)$$

- The reward term is a non-differentiable function of the sample $\boldsymbol{y}$ and the true sequence $\boldsymbol{y}^*$ (e.g., represented by the BLEU metric).

- More generally, objectives of this form are called stochastic objectives:

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim p_\theta(x)}[f(x)]$$

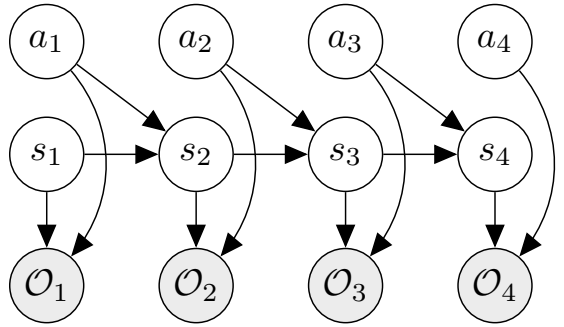
- As we will see, such objectives often come up in RL.



# Plan

## Part 1: Intro to RL and Control as Inference Framework

- Intro to Reinforcement Learning (RL)
- RL and Control as Inference: The GM framework
- Connections to variational inference




## Part 2: Max-entropy RL Algorithms

- Classical Q-learning and policy gradient methods
- Derivation of the soft Q-learning and soft policy gradients
- Algorithms and applications

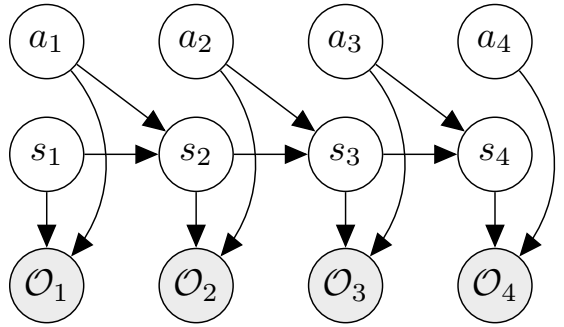
**Algorithm 1** Soft Actor-Critic

Initialize parameter vectors $\psi$, $\bar{\psi}$, $\theta$, $\phi$.
**for** each iteration **do**
  **for** each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  **end for**
  **for** each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  **end for**
**end for**

# Plan

## Part 1: Intro to RL and Control as Inference Framework

- Intro to Reinforcement Learning (RL)
- RL and Control as Inference: The GM framework
- Connections to variational inference




## Part 2: Max-entropy RL Algorithms

- Classical Q-learning and policy gradient methods
- Derivation of the soft Q-learning and soft policy gradients
- Algorithms and applications

**Algorithm 1** Soft Actor-Critic

Initialize parameter vectors $\psi$, $\bar{\psi}$, $\theta$, $\phi$.
**for** each iteration **do**
  **for** each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  **end for**
  **for** each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  **end for**
**end for**

# Paradigms of machine learning

- Supervised learning

  Given: a collection of data $D = \{(x_i, y_i)\}_{i=1}^N$

  Goal: learn a model that approximates $P(y \mid x)$

- Unsupervised learning

  Given: a collection of data $D = \{(x_1, x_2, \dots, x_d)_i\}_{i=1}^N$

  Goal: learn a model that approximates $P(x_1, x_2, \dots, x_d)$

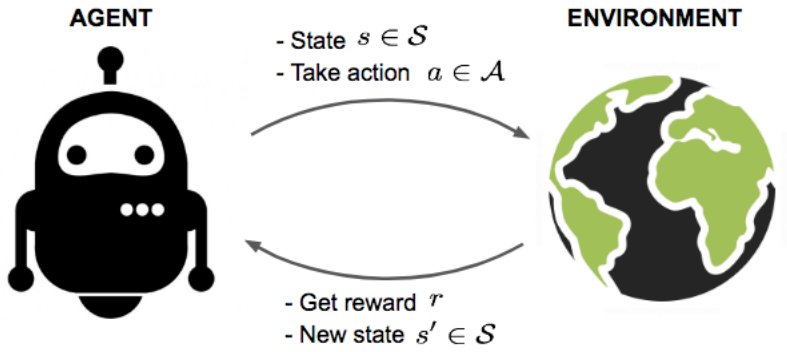
- Reinforcement learning

  Given: an environment that an agent can perceive and interact with

  Goal: learn a controller (policy) that can maximize the utility (reward) in the given environment

GMs allow us to efficiently represent, manipulate, and perform learning and inference on these probabilistic models.

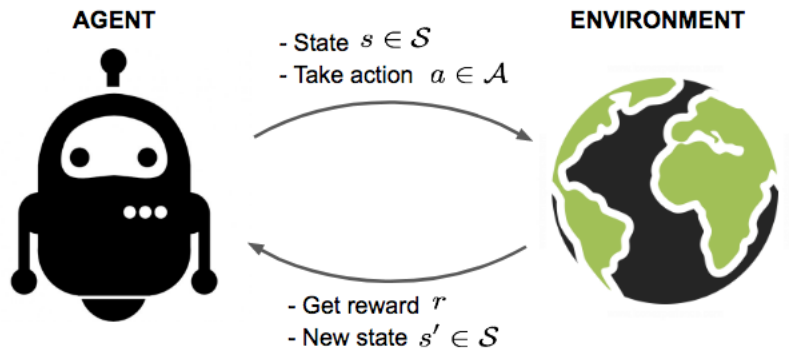
DL gives the tools for learning expressive latent representations that lead to more accurate probabilistic models of the data.



# Why sequential decision making and RL?

Ultimately, we want to build intelligent machines that:

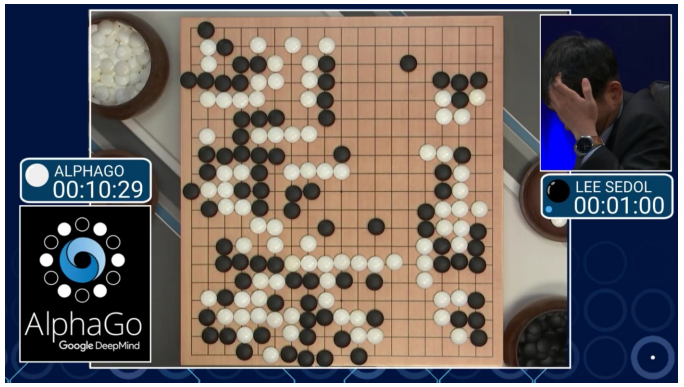
- Can perceive and interact with the world
- Exhibit purposeful goal-directed behavior
- Learn from interactions, adapt to changes, plan and be able to maximize utility functions (specified by humans or inferred from situations)



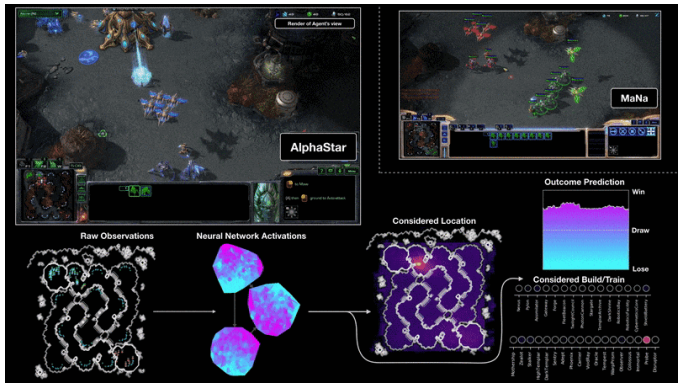
# Some recent success stories of RL

## Learning to play games

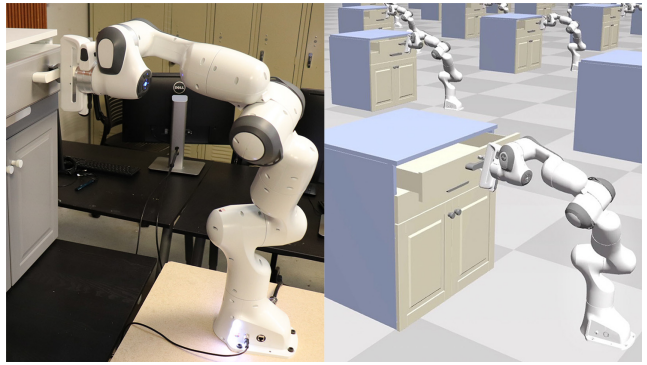
AlphaGo (DeepMind, 2016)



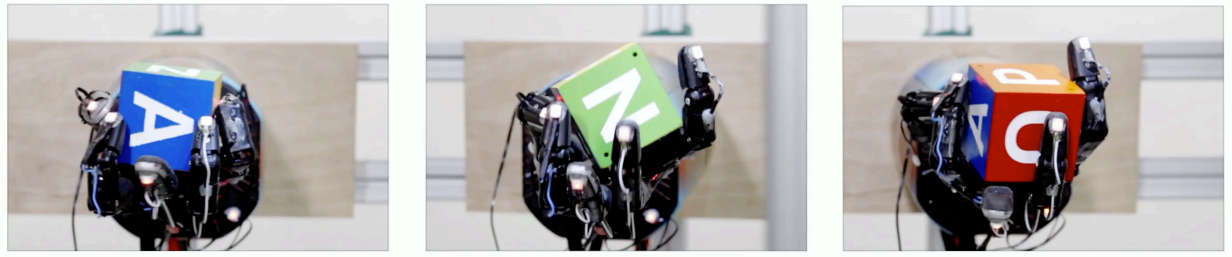

AlphaStar (DeepMind, 2019)




## Robotics



Chebotar et al., 2018




OpenAI, 2018

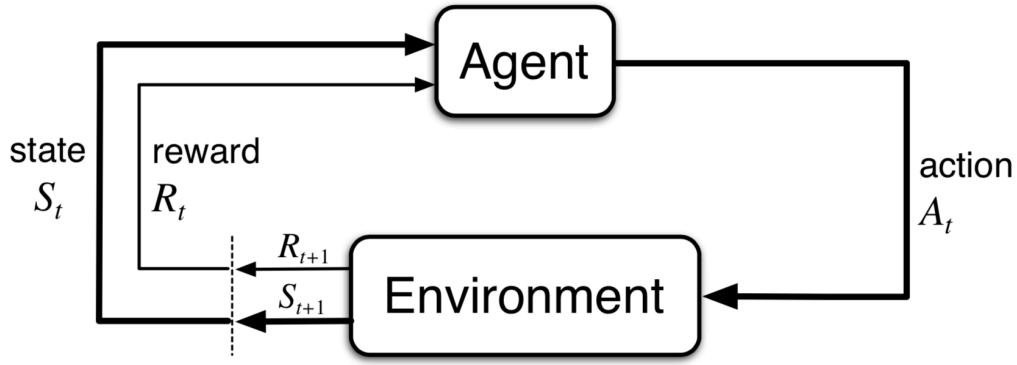
PETUUM

Carnegie Mellon University
School of Computer Science

# Basic concepts of RL

# Markov Decision Processes (MDPs)

Markov Decision Process (MDP):

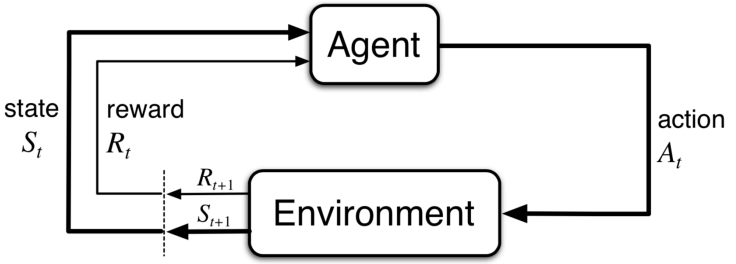



- Environment has a set of states $\mathcal{S}$
- Agent is given a set of possible actions $\mathcal{A}$
- Environment dynamics: transitions from state $s_t$ into a new state $s_{t+1}$ according to the transition probability $P(s_{t+1}|s_t, a_t)$ after agent takes action $a_t$
- Reward function: $\mathrm{r}(s, a) = \mathbb{E}[r_{t+1}|s_t = s, a_t = a]$ provides scalar rewards to the agent at each time step
- "Life" of an agent (or trajectory):

$$\tau = (s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, \dots)$$

Sutton & Barto (2018). Reinforcement learning: An introduction. 2nd edition.

# Markov Decision Processes (MDPs)




What can we do with MDPs:

(1) Policy search: Find a policy $\pi: \mathcal{S} \rightarrow \mathcal{A}$ that outputs actions for each given state such that the cumulative reward along the trajectory is maximized.

(2) Inverse RL: Given a set of optimal trajectories (e.g., generated by a human expert), infer the corresponding MDP.

# Returns and Episodes

Maximization of the return:

- Return (cumulative reward) starting step $t$: $G_t = r_{t+1} + r_{t+2} + \cdots + r_T$
- If $T = \infty$, we can use the notion of discounted return:

$$
\begin{aligned}
G_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \\
&= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \\
&= r_{t+1} + \gamma G_{t+1}
\end{aligned}
$$

where $0 \leq \gamma \leq 1$ is called the discount rate



# Policies and Value Functions

stochastic objective

- Value function of a state $s$:

$$V_\pi(s) := \mathbb{E}_\pi \left[ G_t \mid s_t = s \right] = \mathbb{E}_\pi \left[ \sum_{k=0}^{T} \gamma^k r_{t+k+1} \mid s_t = s \right]$$

- Value function of the state-action pair $(s, a)$:

$$Q_\pi(s, a) := \mathbb{E}_\pi \left[ G_t \mid s_t = s, a_t = a \right]$$

$$= \mathbb{E}_\pi \left[ \sum_{k=0}^{T} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right]$$

# Bellman Equation for $V_\pi(s)$

- Bellman equation for the value function of a state $s$:

$$
\begin{aligned}
V_\pi(s) &:= \mathbb{E}_\pi\left[G_t \mid s_t = s\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{T} \gamma^k r_{t+k+1} \mid s_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma G_{t+1} \mid s_t = s\right] \\
&= \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s, a)\left[r(s,a) + \gamma \mathbb{E}_\pi\left[G_{t+1} \mid s_{t+1} = s'\right]\right] \\
&= \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s, a)\left[r(s,a) + \gamma V_\pi(s')\right]
\end{aligned}
$$



# Bellman Equation for $V_\pi(s)$

- Bellman equation for the value function of a state $s$:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s, a) \left[ r(s, a) + \gamma V_\pi(s') \right]$$

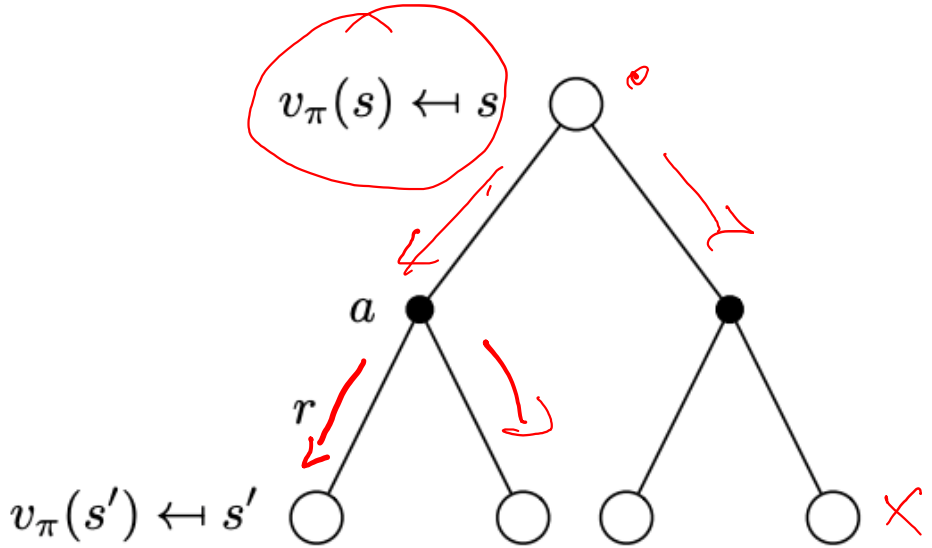
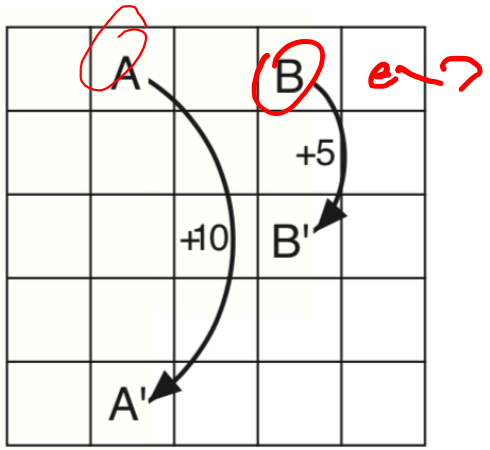
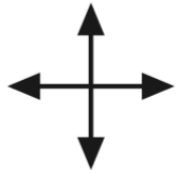



Illustration from David Silver's lecture.

# Example: Grid World and a Random Policy

- Getting off the grid results in -1 reward and no change in position.
- Any action in A and B result in +10 and +5 and move the agent to A' and B', respectively.




| | | | | |
|---|---|---|---|---|
| 3.3 | 8.8 | 4.4 | 5.3 | 1.5 |
| 1.5 | 3.0 | 2.3 | 1.9 | 0.5 |
| 0.1 | 0.7 | 0.7 | 0.4 | -0.4 |
| -1.0 | -0.4 | -0.4 | -0.6 | -1.2 |
| -1.9 | -1.3 | -1.2 | -1.4 | -2.0 |

**Figure 3.2:** Gridworld example: exceptional reward dynamics (left) and state-value function for the equiprobable random policy (right).

# Bellman Equation for $Q_\pi(s, a)$

- Bellman equation for the value function of the state-action pair $(s, a)$:

$$Q_\pi(s,a) := \mathbb{E}_\pi \left[ G_t \mid s_t = s, a_t = a \right] = \mathbb{E}_\pi \left[ \sum_{k=0}^{T} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right]$$

$$= r(s,a) + \gamma \mathbb{E}_\pi \left[ G_{t+1} \mid s_t = s, a_t = a \right]$$

$$= r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \sum_{a'} \pi(a' \mid s') \mathbb{E}_\pi \left[ G_{t+1} \mid s_{t+1} = s', a_{t+1} = a' \right]$$

$$= r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \sum_{a'} \pi(a' \mid s') Q_\pi(s', a')$$



# Bellman Equation for $Q_\pi(s, a)$

- Bellman equation for the value function of the state-action pair $(s, a)$:

$$Q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \sum_{a'} \pi(a' \mid s') Q_\pi(s',a')$$

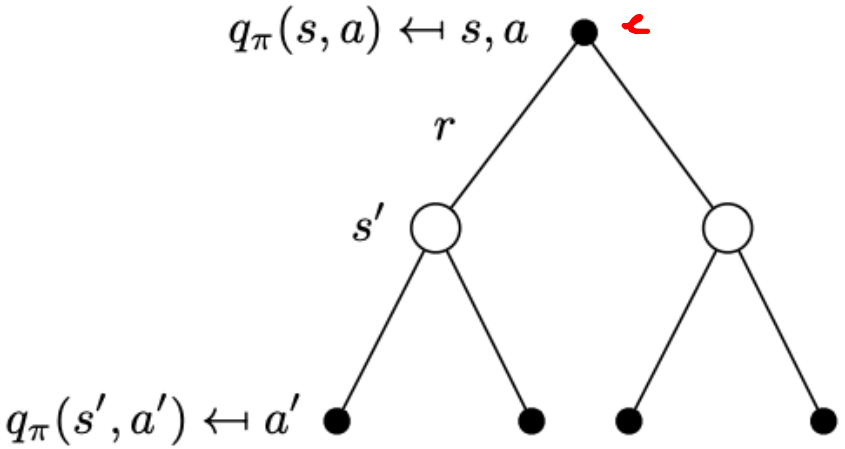



Illustration from David Silver's lecture.



# Optimal Policies and Value Functions

- Solving an RL task means finding an optimal policy that achieves high reward in the long run.
- Policy $\pi$ is better or equal to $\pi'$ ($\pi \geq \pi'$) if its expected return is greater or equal to that of $\pi'$ for all states:

$$\pi \geq \pi' \Leftrightarrow V_\pi(s) \geq V_{\pi'}(s)\ \forall s \in \mathcal{S}$$

- Optimal value functions (Bellman optimality):

$$V_\star(s) := \max_\pi V_\pi(s) = \max_a \sum_{s'} p(s' \mid s, a)\left[r(s,a) + \gamma V_\star(s')\right]$$

$$Q_\star(s,a) := \max_\pi Q_\pi(s,a) = \sum_{s'} p(s' \mid s, a)\left[r(s,a) + \gamma \max_{a'} Q_\star(s', a')\right]$$



# Optimal Policies and Value Functions

- Optimal value functions (Bellman optimality):

$$V_\star(s) := \max_\pi V_\pi(s) = \max_a \sum_{s'} p(s' \mid s, a) \left[ r(s, a) + \gamma V_\star(s') \right]$$

$$Q_\star(s, a) := \max_\pi Q_\pi(s, a) = \sum_{s'} p(s' \mid s, a) \left[ r(s, a) + \gamma \max_{a'} Q_\star(s', a') \right]$$

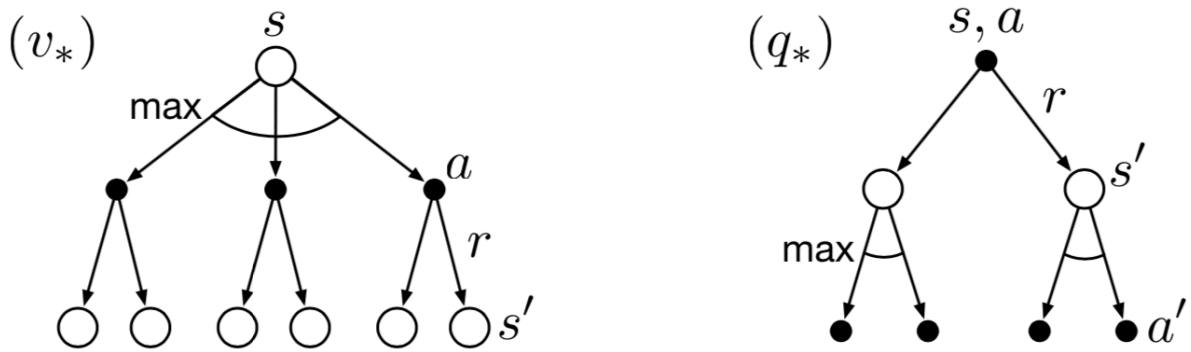



**Figure 3.4:** Backup diagrams for $v_*$ and $q_*$

Sutton & Barto (2018). Reinforcement learning: An introduction. 2nd edition.

# How to recover optimal policy and trajectories?

- We can recover an optimal policy from the optimal $Q_*\ (s, a)$:

$$\pi_\star(a \mid s) = \delta\left(a = \arg\max_a Q_\star(s, a)\right)$$

Dirac delta

- To recover a set of optimal trajectories, just execute the optimal policy:

$$\tau_\star = (s_1^\star, a_1^\star, r_1^\star, s_2^\star, a_2^\star, r_2^\star, \dots)$$

$$s_{t+1}^\star \sim p(s_{t+1} \mid s_t, a_t^\star = \arg\max_a Q_\star(s, a))$$



# Example: Grid World and an Optimal Policy

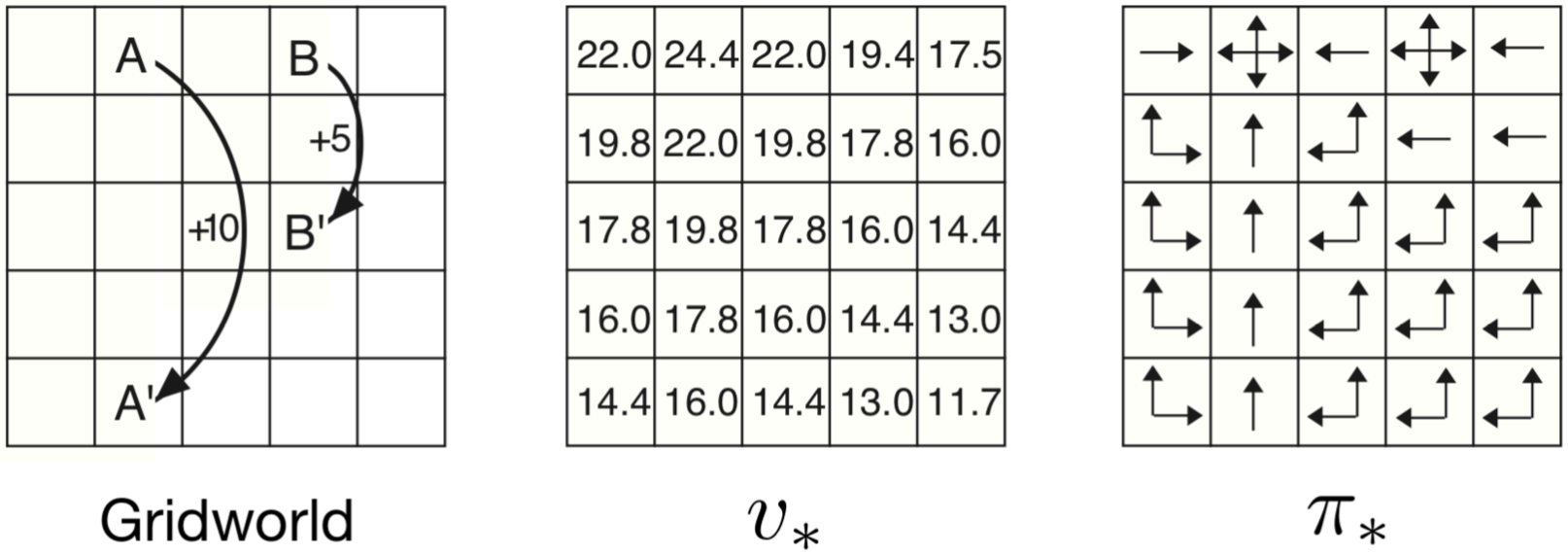
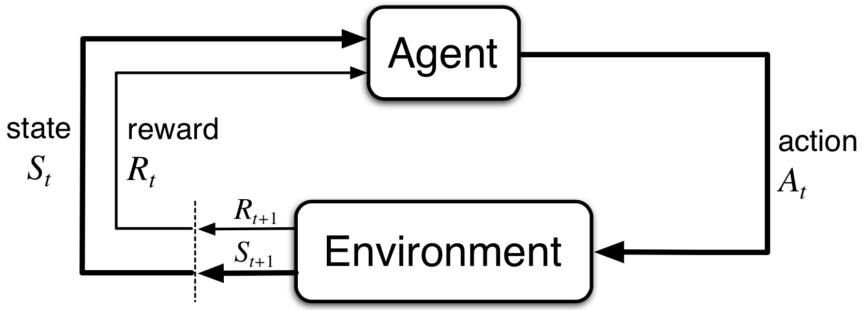



**Figure 3.5:** Optimal solutions to the gridworld example.

Sutton & Barto (2018). Reinforcement learning: An introduction. 2nd edition.

# Recap




| | |
|---|---|
| Initial state | $s_0 \sim p_0(s)$ |
| Transition | $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$ |
| Policy | $a_t \sim \pi(a_t \mid s_t)$ |
| Reward | $r_t = r(s_t, a_t)$ |

- Value functions:

$$V_\pi(s) := \mathbb{E}_\pi \left[ \sum_{k=0}^{T} \gamma^k r_{t+k+1} \mid s_t = s \right]$$

$$Q_\pi(s,a) := \mathbb{E}_\pi \left[ \sum_{k=0}^{T} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right]$$

- Recursive notion of optimality:

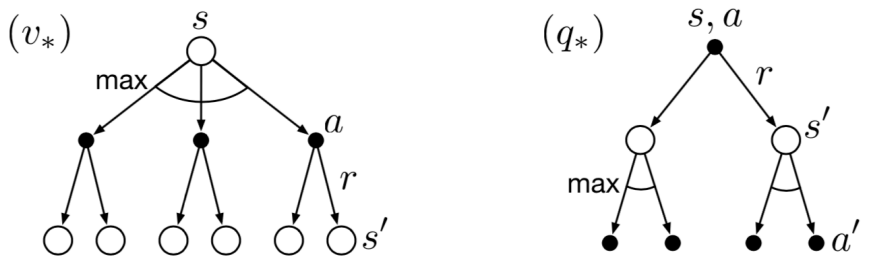


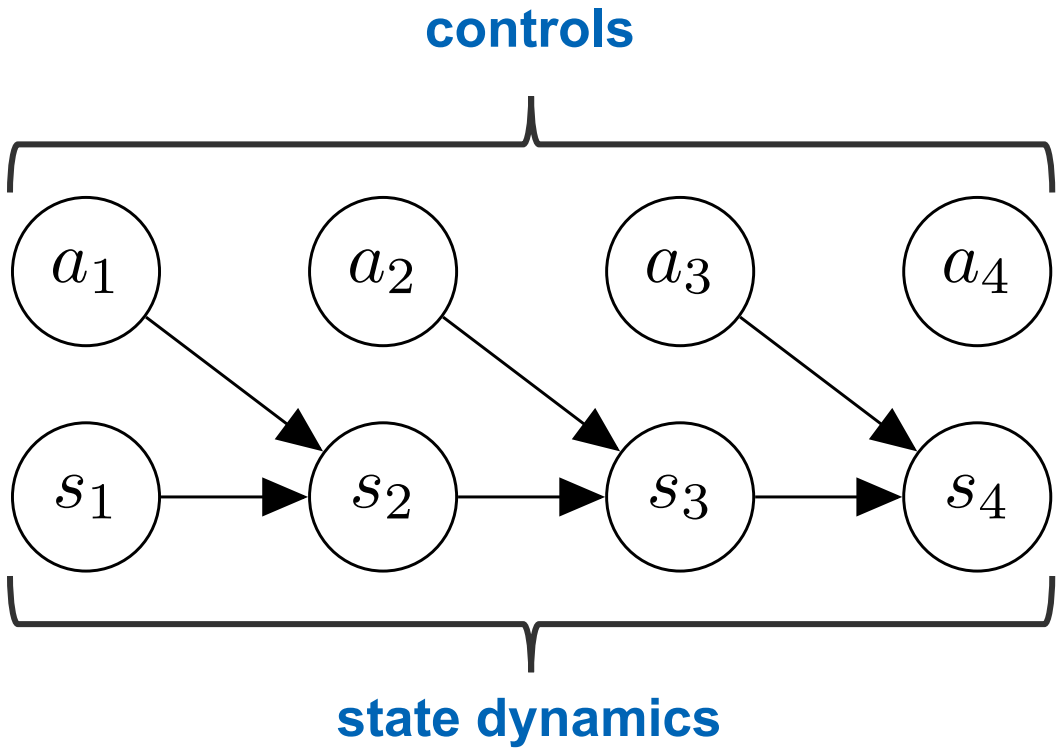

**Figure 3.4:** Backup diagrams for $v_*$ and $q_*$

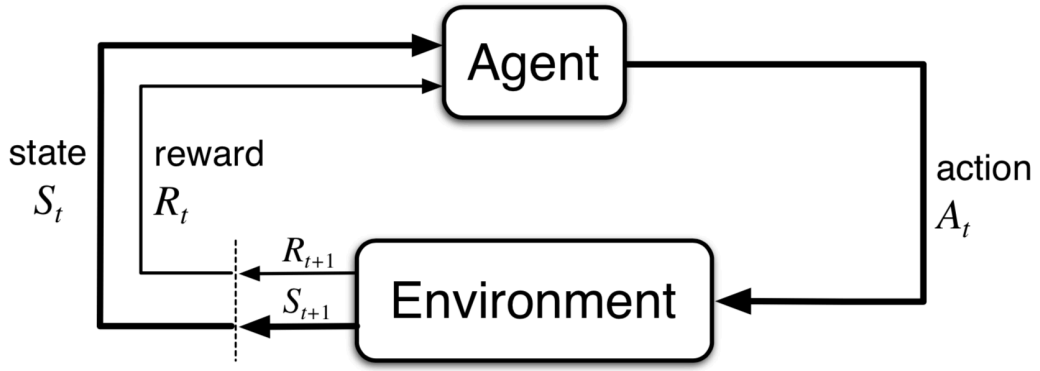
PETUUM

Carnegie Mellon University
School of Computer Science

# RL & Control as Inference in GM

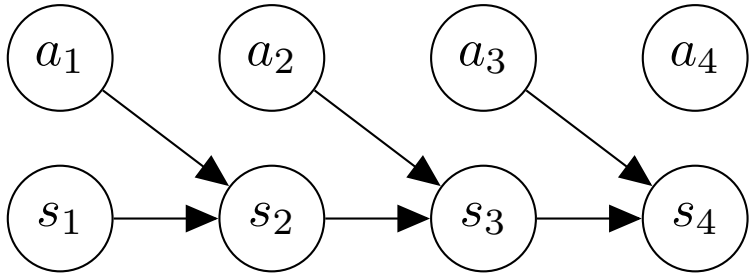
# MDP as a Graphical Model



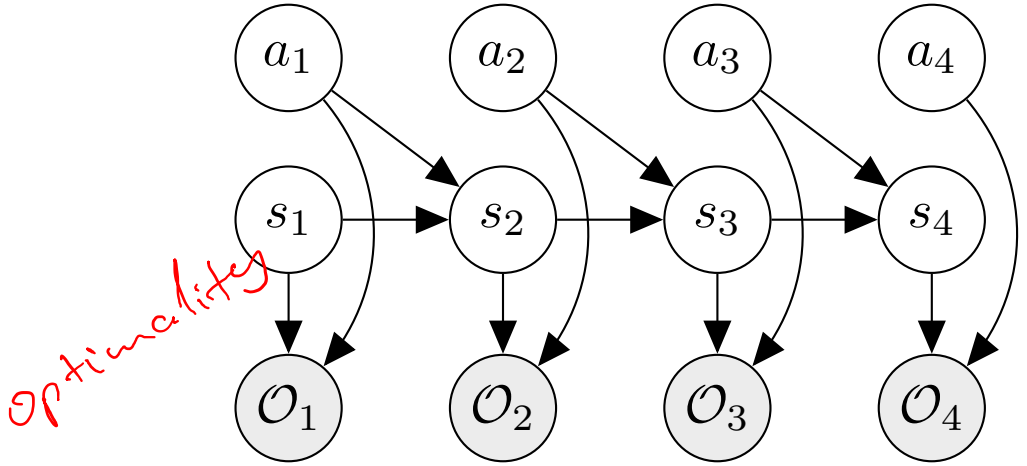




**How do we define a distribution over rational/optimal trajectories?**

# MDP as a Graphical Model

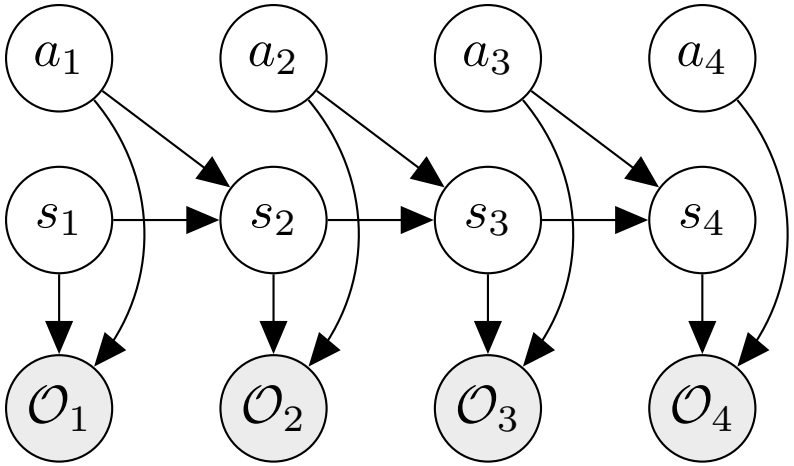



| | |
|---|---|
| Initial state | $s_0 \sim p_0(s)$ |
| Transition | $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$ |
| Policy | $a_t \sim \pi(a_t \mid s_t)$ |
| Reward | $r_t = r(s_t, a_t)$ |



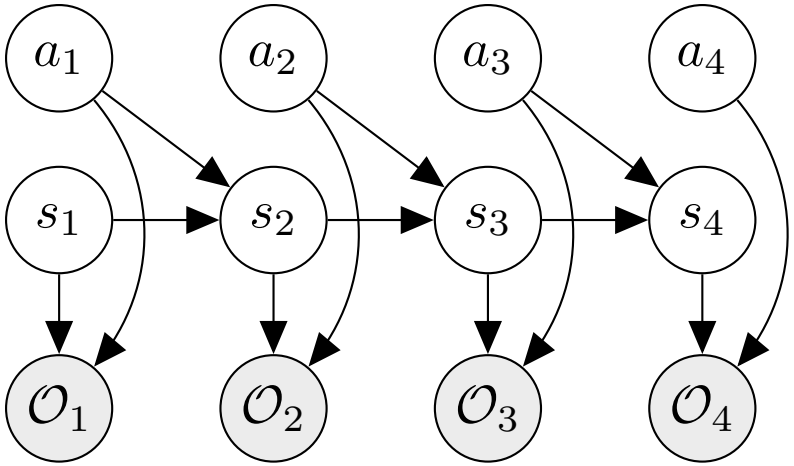

| | |
|---|---|
| Initial state | $s_0 \sim p_0(s)$ |
| Transition | $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$ |
| Policy | $a_t \sim \pi(a_t \mid s_t)$ |
| Reward | $r_t = r(s_t, a_t)$ |
| Optimality | $p(\mathcal{O}_t = 1 \mid s_t, a_t) = \exp(r(s_t, a_t))$ |

# Why is this model interesting?




- Can solve control and planning problems using inference algorithms
- Allows to model suboptimal behavior (important for inverse RL)
- Provides and explanation for why stochastic behavior might be preferred (from the exploration and transfer learning point of view)

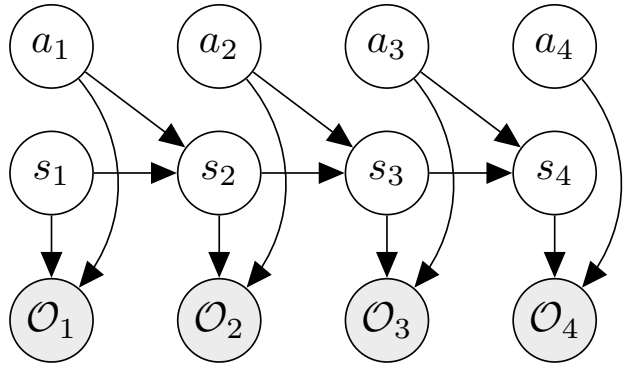
# What can we do with this graphical model?




Here is what we can do:

- Given a reward, determine a likely optimal trajectory $p(\tau \mid \mathcal{O}_{1:T})$
- Given a collection of optimal trajectories, infer the reward and priors
- Given a reward, infer the optimal policy $p(a_t \mid s_t, \mathcal{O}_{t:T})$

# Distribution over the optimal trajectories




$$p(\mathcal{O}_t \mid s_t, a_t) = \exp(r(s_t, a_t))$$

$$p(\tau \mid \mathcal{O}_{1:T}) \propto p(s_1) \prod_{t=1}^{T} p(a_t \mid s_t) p(s_{t+1} \mid s_t, a_t) p(\mathcal{O}_t \mid s_t, a_t)$$

action prior

Levine (2018). Reinforcement learning and control as probabilistic inference.

# Inferring the reward & prior that generate trajectories

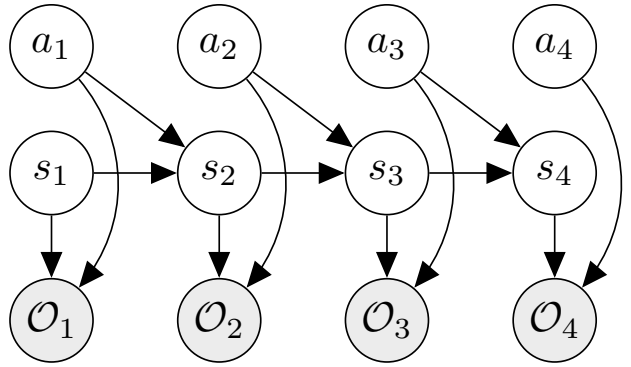
$$p(\tau \mid \mathcal{O}_{1:T}, \theta, \phi) \propto \left[ p(s_1) \prod_{t=1}^{T} p(s_{t+1} \mid s_t, a_t) \right] \exp\left( \sum_{t=1}^{T} r_\phi(s_t, a_t) + \log p_\theta(a_t \mid s_t) \right)$$

The model reminds a featurized CRF. (*Note: CRF is undirected and does not preserve the causal structure; this model is more restrictive and known as MEMM.)

Ziebart (2010). Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy. PhD thesis.



# Optimal policy and planning via inference




- Unroll the dynamics and compute backward messages:

$$\beta_t(s_t, a_t) := p(\mathcal{O}_{t:T} \mid s_t, a_t)$$

- Compute optimal policy:

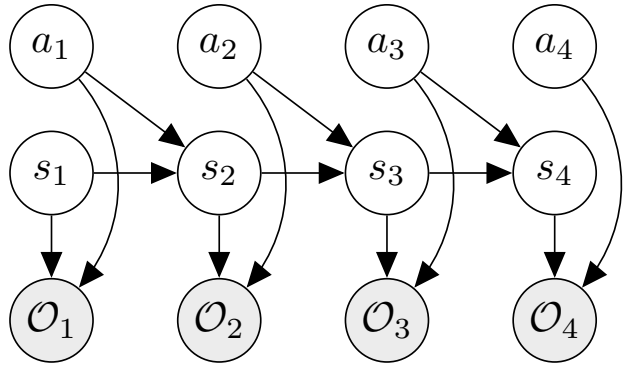
$$p(a_t \mid s_t, \mathcal{O}_{t:T})$$

- Compute forward messages (state filtering under optimality constraint):

$$\alpha_t(s_t) = p(s_t \mid \mathcal{O}_{1:t-1})$$

Levine (2018). Reinforcement learning and control as probabilistic inference.

# Backward messages




$$\beta_t(s_t, a_t) = p(\mathcal{O}_{t:T} \mid s_t, a_t)$$

**Probability that we can be optimal at steps $t$ through $T$ given $s_t$ and $a_t$**

$$= \int_{\mathcal{S}} p(\mathcal{O}_{t:T}, s_{t+1} \mid s_t, a_t) ds_{t+1}$$

$$= \int_{\mathcal{S}} p(\mathcal{O}_{t+1:T} \mid s_{t+1}) p(s_{t+1} \mid s_t, a_t) p(\mathcal{O}_t \mid s_t, a_t) ds_{t+1}$$

$$\underbrace{p(\mathcal{O}_{t+1:T} \mid s_{t+1})}_{\beta_{t+1}(s_{t+1})} = \int_{\mathcal{A}} p(\mathcal{O}_{t+1:T} \mid s_{t+1}, a_{t+1}) p(a_{t+1} \mid s_{t+1}) da_{t+1}$$

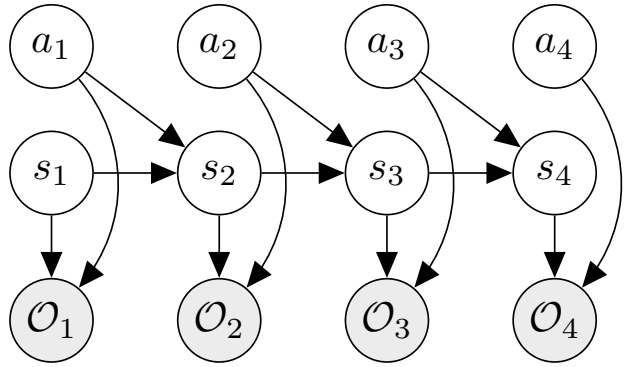
for $t = T - 1$ to $1$ :

$$\beta_t(s_t, a_t) = p(\mathcal{O}_t \mid s_t, a_t) \mathbb{E}_{s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)} \left[ \beta_{t+1}(s_{t+1}) \right]$$

$$\beta_t(s_t) = \mathbb{E}_{a_t \sim p(a_t \mid s_t)} \left[ \beta_t(s_t, a_t) \right]$$

Levine (2018). Reinforcement learning and control as probabilistic inference.

# How are these messages related to RL?




$$\text{let } V_t(\mathbf{s}_t) = \log \beta_t(\mathbf{s}_t)$$

$$\text{let } Q_t(\mathbf{s}_t, \mathbf{a}_t) = \log \beta_t(\mathbf{s}_t, \mathbf{a}_t)$$

$$V(\mathbf{s}_t) = \log \int \exp(Q(\mathbf{s}_t, \mathbf{a}_t) + \log p(\mathbf{a}_t|\mathbf{s}_t))\mathbf{a}_t$$

softmax

**Deterministic dynamics:** $Q(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + V(\mathbf{s}_{t+1})$

$\approx$ logsumexp

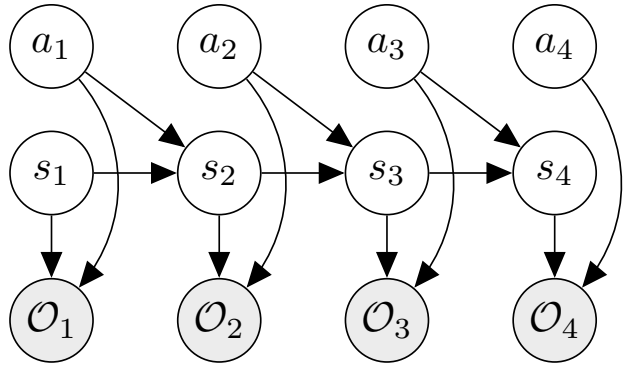
$\approx$ softmax

**Stochastic dynamics:** $Q(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \log E_{\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)}[\exp(V(\mathbf{s}_{t+1}))]$

**"optimistic" transition (not good)**

Levine (2018). Reinforcement learning and control as probabilistic inference.

# Optimal policy



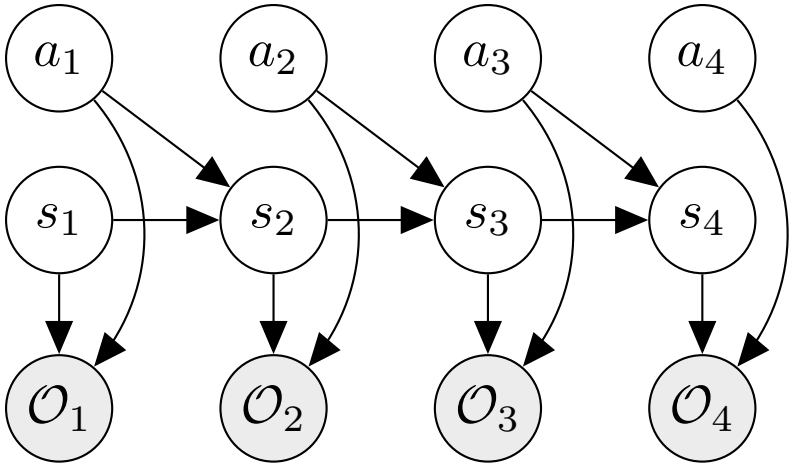

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \frac{\beta_t(\mathbf{s}_t, \mathbf{a}_t)}{\beta_t(\mathbf{s}_t)} \quad | \; p(a_t | s_t, \theta_{t:T})$$

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t) - V_t(\mathbf{s}_t)) = \exp(A_t(\mathbf{s}_t, \mathbf{a}_t)) \quad \frac{1}{\alpha})$$

- (Derivation: HW4!)
- Natural interpretation: better actions are more probable + random tie breaking
- Approaches greedy policy as temperature decreases

Levine (2018). Reinforcement learning and control as probabilistic inference.

# Summary




- Using auxiliary potentials and/or optimality variables, we reduced optimal control to inference in a graphical model.
- “Solving MDP” becomes very similar to inference in HMM / MEMM / CRF.
- The approach is quite similar to dynamic programming, value iterations, etc.

PETUUM

Carnegie Mellon University
School of Computer Science

# Control via Variational Inference

# Which objective does inference optimize?

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \frac{\beta_t(\mathbf{s}_t, \mathbf{a}_t)}{\beta_t(\mathbf{s}_t)}$$

?

$$\pi_\star = \arg\max_\pi \mathbb{E}\left[V_\pi(s)\right]$$

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t) - V_t(\mathbf{s}_t)) = \exp(A_t(\mathbf{s}_t, \mathbf{a}_t))$$

- Q: Is there a way to find an objective function by optimizing which we can recover our inference-based policy?
- A: Yes! Let's take a look at the KL divergence between trajectory distributions.



# Which objective does inference optimize?

## [the case of deterministic dynamics]

**Optimal**

$$p(\tau) = \left[ p(\mathbf{s}_1) \prod_{t=1}^{T} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \right] \exp\left( \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right)$$

$$-D_{\mathrm{KL}}(\hat{p}(\tau)\|p(\tau)) = E_{\tau\sim\hat{p}(\tau)}\left[ \log p(\mathbf{s}_1) + \sum_{t=1}^{T} (\log p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) + r(\mathbf{s}_t, \mathbf{a}_t)) - \right.$$
$$\left. \log p(\mathbf{s}_1) - \sum_{t=1}^{T} (\log p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) + \log \pi(\mathbf{a}_t|\mathbf{s}_t)) \right]$$

# Which objective does inference optimize?

## [the case of deterministic dynamics]

under policy $\pi(a|s)$

$$-D_{\mathrm{KL}}(\hat{p}(\tau)\|p(\tau)) = E_{\tau\sim\hat{p}(\tau)}\left[\sum_{t=1}^{T} r(\mathbf{s}_t,\mathbf{a}_t) - \log\pi(\mathbf{a}_t|\mathbf{s}_t)\right]$$

$$= \sum_{t=1}^{T} E_{(\mathbf{s}_t,\mathbf{a}_t)\sim\hat{p}(\mathbf{s}_t,\mathbf{a}_t))}[r(\mathbf{s}_t,\mathbf{a}_t) - \log\pi(\mathbf{a}_t|\mathbf{s}_t)]$$

$$= \sum_{t=1}^{T} E_{(\mathbf{s}_t,\mathbf{a}_t)\sim\hat{p}(\mathbf{s}_t,\mathbf{a}_t))}[r(\mathbf{s}_t,\mathbf{a}_t)] + E_{\mathbf{s}_t\sim\hat{p}(\mathbf{s}_t)}[\mathcal{H}(\pi(\mathbf{a}_t|\mathbf{s}_t))]$$

entropy of the policy

expected returns

standard RL

"given that you obtained high reward, what was your transition probability?"

# The problem of optimism in stochastic dynamics

**Deterministic dynamics:** $Q(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + V(\mathbf{s}_{t+1})$

**Stochastic dynamics:** $Q(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \log E_{\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)}[\exp(V(\mathbf{s}_{t+1}))]$

**"optimistic" transition (not good)**

**Why did this happen?**

- Dichotomy between what resulted in the high reward:
  was it a good policy or we just got lucky with the stochastic dynamics?
- The optimal policy: $p(a_t \mid s_t, \mathcal{O}_{t:T})$

"given that you obtained high reward, what was your action probability?"

- The "optimal" transition probability: $p(s_{t+1} \mid s_t, a_t, \mathcal{O}_{t:T}) \neq p(s_{t+1} \mid s_t, a_t)$

"given that you obtained high reward, what was your transition probability?"

Adapted from Sergey Levine's slides.

"given that you obtained high reward, what was your transition probability?"

# Control via variational inference

"given that you obtained high reward, what was your action probability given that your transition probability did not change?"

Let's find $q(s_{1:T}, a_{1:T})$ ← $q(\tau)$

such that it approximates $p(s_{1:T}, a_{1:T} \mid \mathcal{O}_{1:T})$ ← $p(\tau \mid \mathcal{O}_{1:T})$

while the dynamics stays fixed to $p(s_{t+1} \mid s_t, a_t)$

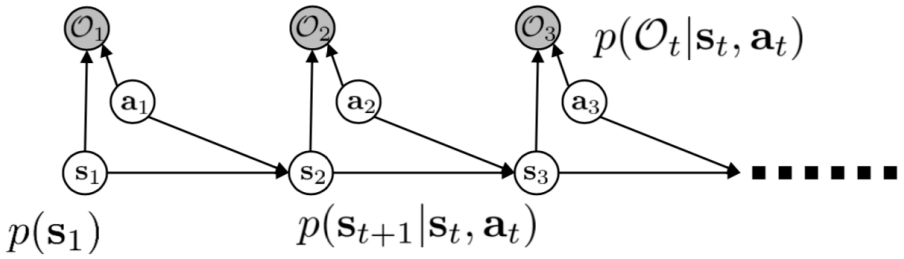
Adapted from Sergey Levine's slides.



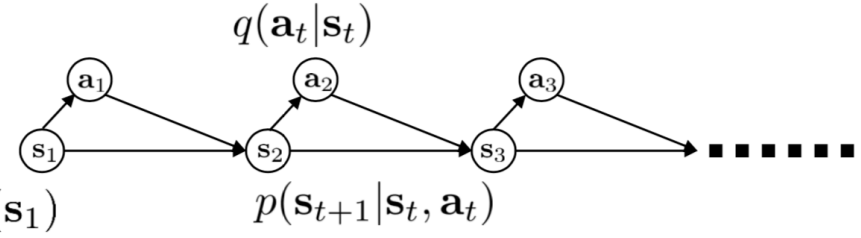
# Control via variational inference

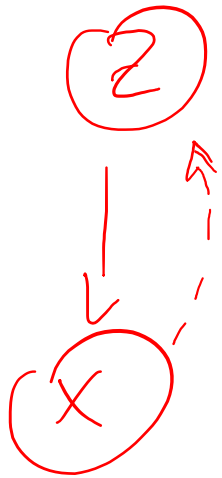
$$\text{let } q(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}) = p(\mathbf{s}_1) \prod_t p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) q(\mathbf{a}_t|\mathbf{s}_t)$$

$p(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}|\mathcal{O}_{1:T})$



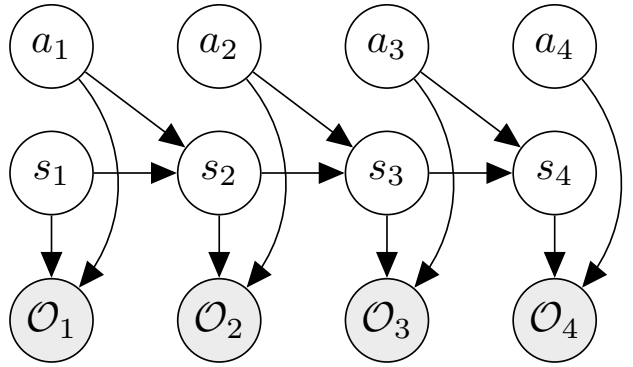

$q(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})$



Adapted from Sergey Levine's slides.

# Control via variational inference




**Optimal**

$$p(\tau) = \left[p(\mathbf{s}_1)\prod_{t=1}^{T} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\right] \exp\left(\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\right)$$

**Produced by a policy**

$$q(\tau) = q(\mathbf{s}_1)\prod_{t=1}^{T} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)q(\mathbf{a}_t|\mathbf{s}_t)$$

policy

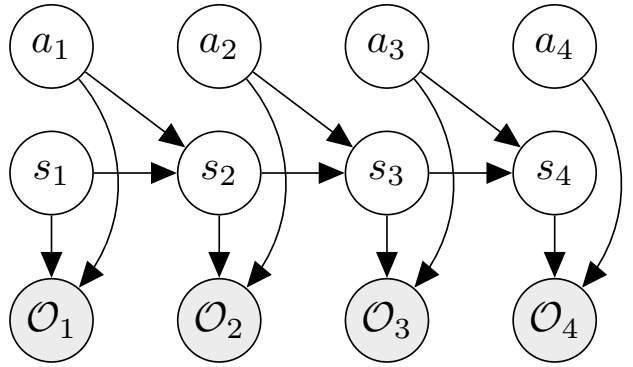
$$\log p(\mathcal{O}_{1:T}) = \log \int\int p(\mathcal{O}_{1:T}, \mathbf{s}_{1:T}, \mathbf{a}_{1:T}) d\mathbf{s}_{1:T} d\mathbf{a}_{1:T}$$

$$= \log \int\int p(\mathcal{O}_{1:T}, \mathbf{s}_{1:T}, \mathbf{a}_{1:T}) \frac{q(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})}{q(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})} d\mathbf{s}_{1:T} d\mathbf{a}_{1:T}$$

Jensen's

ELBO

# Control via variational inference




$$\log p(\mathcal{O}_{1:T}) \geq \mathbb{E}_{\tau\sim q}\Bigg[\log p(s_1) + \sum_{t=1}^{T}\log p(s_{t+1}\mid s_t, a_t) + \sum_{t=1}^{T}\log p(\mathcal{O}_t\mid s_t, a_t)$$

$$- \log p(s_1) - \sum_{t=1}^{T}\log p(s_{t+1}\mid s_t, a_t) - \sum_{t=1}^{T}\log q(a_t\mid s_t)\Bigg]$$

$$\geq \mathbb{E}_{\tau\sim q}\left[\sum_{t=1}^{T} r(s_t, a_t) - \log q(a_t\mid s_t)\right]$$

$$= \sum_{t=1}^{T}\mathbb{E}_{(s_t,a_t)\sim q}\left[r(s_t, a_t)\right] + \mathcal{H}(q(a_t\mid s_t))$$

expected rewards

entropy of the actions.

# Optimal policy with respect to ELBO

let $q(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}) = p(\mathbf{s}_1) \prod_t p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) q(\mathbf{a}_t|\mathbf{s}_t)$

$$\log p(\mathcal{O}_{1:T}) \geq \sum_t E_{(\mathbf{s}_t, \mathbf{a}_t) \sim q} \left[ r(\mathbf{s}_t, \mathbf{a}_t) + \mathcal{H}(q(\mathbf{a}_t|\mathbf{s}_t)) \right]$$

base case: solve for $q(\mathbf{a}_T|\mathbf{s}_T)$:

$$q(\mathbf{a}_T|\mathbf{s}_T) = \arg\max E_{\mathbf{s}_T \sim q(\mathbf{s}_T)} \left[ E_{\mathbf{a}_T \sim q(\mathbf{a}_T|\mathbf{s}_T)} [r(\mathbf{s}_T, \mathbf{a}_T)] + \mathcal{H}(q(\mathbf{a}_T|\mathbf{s}_T)) \right]$$

$$= \arg\max E_{\mathbf{s}_T \sim q(\mathbf{s}_T)} \left[ E_{\mathbf{a}_T \sim q(\mathbf{a}_T|\mathbf{s}_T)} [r(\mathbf{s}_T, \mathbf{a}_T) - \log q(\mathbf{a}_T|\mathbf{s}_T)] \right]$$

minimized when $q(\mathbf{a}_T|\mathbf{s}_T) \propto \exp(r(\mathbf{s}_T, \mathbf{a}_T))$

$$q(\mathbf{a}_T|\mathbf{s}_T) = \frac{\exp(r(\mathbf{s}_T, \mathbf{a}_T))}{\int \exp(r(\mathbf{s}_T, \mathbf{a}))d\mathbf{a}} = \exp(Q(\mathbf{s}_T, \mathbf{a}_T) - V(\mathbf{s}_T)) \qquad V(\mathbf{s}_T) = \log \int \exp(Q(\mathbf{s}_T, \mathbf{a}_T))d\mathbf{a}_T$$

$$E_{\mathbf{s}_T \sim q(\mathbf{s}_T)} \left[ E_{\mathbf{a}_T \sim q(\mathbf{a}_T|\mathbf{s}_T)} [r(\mathbf{s}_T, \mathbf{a}_T) - \log q(\mathbf{a}_T|\mathbf{s}_T)] \right] = E_{\mathbf{s}_T \sim q(\mathbf{s}_T)} \left[ E_{\mathbf{a}_T \sim q(\mathbf{a}_T|\mathbf{s}_T)} [V(\mathbf{s}_T)] \right]$$

# Optimal policy with respect to ELBO

$$\log p(\mathcal{O}_{1:T}) \geq \sum_t E_{(\mathbf{s}_t,\mathbf{a}_t)\sim q}\left[r(\mathbf{s}_t,\mathbf{a}_t) + \mathcal{H}(q(\mathbf{a}_t|\mathbf{s}_t))\right]$$

$$q(\mathbf{a}_T|\mathbf{s}_T) = \frac{\exp(r(\mathbf{s}_T,\mathbf{a}_T))}{\int \exp(r(\mathbf{s}_T,\mathbf{a}))d\mathbf{a}} = \exp(Q(\mathbf{s}_T,\mathbf{a}_T) - V(\mathbf{s}_T))$$

$$E_{\mathbf{s}_T\sim q(\mathbf{s}_T)}\left[E_{\mathbf{a}_T\sim q(\mathbf{a}_T|\mathbf{s}_T)}[r(\mathbf{s}_T,\mathbf{a}_T) - \log q(\mathbf{a}_T|\mathbf{s}_T)]\right] = E_{\mathbf{s}_T\sim q(\mathbf{s}_T)}\left[E_{\mathbf{a}_T\sim q(\mathbf{a}_T|\mathbf{s}_T)}[V(\mathbf{s}_T)]\right]$$

$$\begin{aligned} q(\mathbf{a}_t|\mathbf{s}_t) &= \arg\max E_{\mathbf{s}_t\sim q(\mathbf{s}_t)}\left[E_{\mathbf{a}_t\sim q(\mathbf{a}_t|\mathbf{s}_t)}[r(\mathbf{s}_t,\mathbf{a}_t) + E_{\mathbf{s}_{t+1}\sim p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)}[V(\mathbf{s}_{t+1})]] + \mathcal{H}(q(\mathbf{a}_t|\mathbf{s}_t))\right] \\ &= \arg\max E_{\mathbf{s}_t\sim q(\mathbf{s}_t)}\left[E_{\mathbf{a}_t\sim q(\mathbf{a}_t|\mathbf{s}_t)}[Q(\mathbf{s}_t,\mathbf{a}_t)] + \mathcal{H}(q(\mathbf{a}_t|\mathbf{s}_t))\right] \\ &= \arg\max E_{\mathbf{s}_t\sim q(\mathbf{s}_t)}\left[E_{\mathbf{a}_t\sim q(\mathbf{a}_t|\mathbf{s}_t)}[Q(\mathbf{s}_t,\mathbf{a}_t) - \log q(\mathbf{a}_t|\mathbf{s}_t)]\right] \end{aligned}$$

minimized when $q(\mathbf{a}_t|\mathbf{s}_t) \propto \exp(Q(\mathbf{s}_t,\mathbf{a}_t))$

$$V_t(\mathbf{s}_t) = \log \int \exp(Q_t(\mathbf{s}_t,\mathbf{a}_t))d\mathbf{a}_t$$

$$q(\mathbf{a}_t|\mathbf{s}_t) = \exp(Q(\mathbf{s}_t,\mathbf{a}_t) - V(\mathbf{s}_t))$$

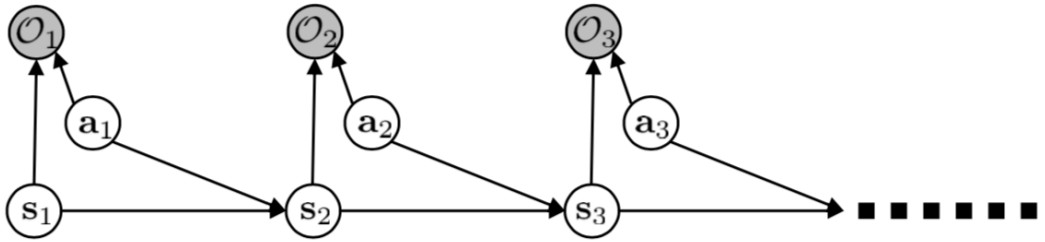
$$Q_t(\mathbf{s}_t,\mathbf{a}_t) = r(\mathbf{s}_t,\mathbf{a}_t) + E[(V_{t+1}(\mathbf{s}_{t+1})]$$

*regular* Bellman backup

*not* optimistic

# Summary

$p(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}|\mathcal{O}_{1:T})$




$q(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})$


$$V_t(\mathbf{s}_t) = \log \int \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t))d\mathbf{a}_t \qquad Q_t(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + E[(V_{t+1}(\mathbf{s}_{t+1})]$$

PETUUM

Carnegie Mellon University
School of Computer Science