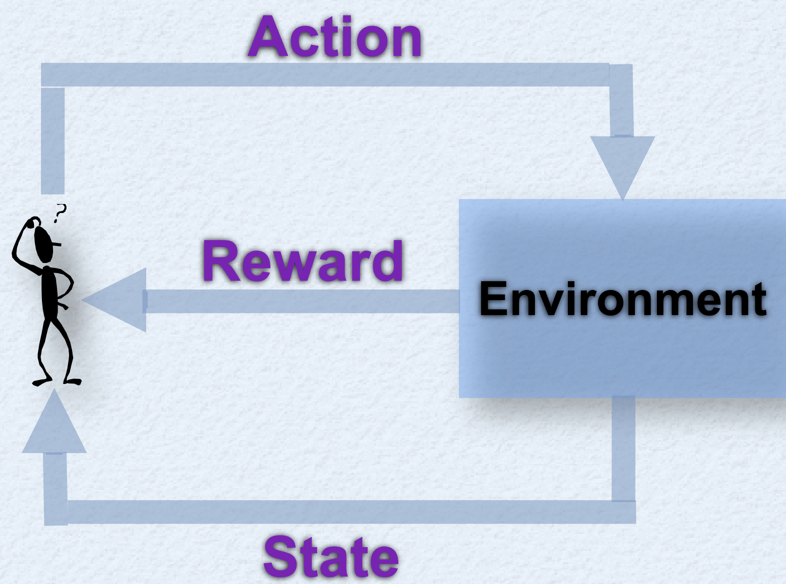
Introduction to Reinforcement Learning

Sequential Decision Making under Uncertainty

How Can I ... ?

- Move around in the physical world **(e.g. driving, navigation)**
- Play and win a game
- Retrieve information over the web
- Do medical diagnosis and treatment
- Maximize the throughput of a factory
- Optimize the performance of a rescue team

Reinforcement Learning



- **RL:** A class of learning problems in which an agent interacts with an unfamiliar, dynamic and stochastic environment
- **Goal:** Learn a policy to maximize some measure of long-term reward
- **Interaction:** Modeled as a MDP or a POMDP

Bayes RL

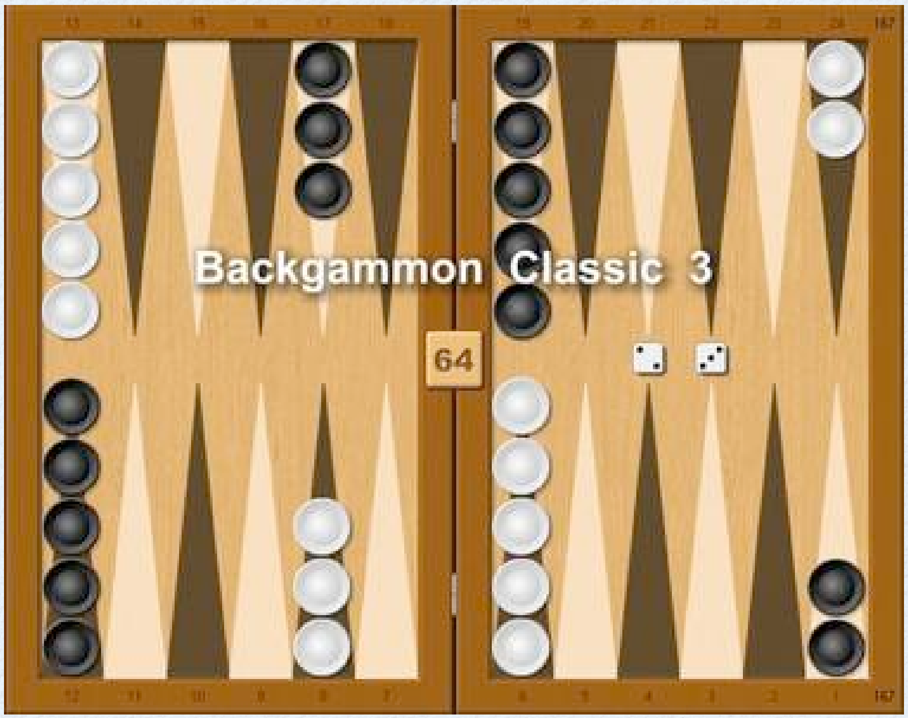
Markov Decision Processes

- **An MDP is defined as a 5-tuple** $(\mathcal{X}, \mathcal{A}, p, q, p_0)$
  - $\mathcal{X}$: State space of the process
  - $\mathcal{A}$: Action space of the process
  - $p(\cdot|x,a)$: Probability distribution over next state $x_{t+1} \sim p(\cdot|x_t, a_t)$
  - $q(\cdot|x,a)$: Probability distribution over rewards $R(x_t, a_t) \sim q(\cdot|x_t, a_t)$
  - p_0 : Initial state distribution

- **Policy:** Mapping from states to actions or distributions over actions

$$\mu(x) \in \mathcal{A} \qquad \text{or} \qquad \mu(\cdot|x) \in \Pr(\mathcal{A})$$

Example: Backgammon



- **States:** board configurations (about 10^{20})
- **Actions:** permissible moves
- **Rewards:** win +1, lose -1, else 0

RL Applications

- Backgammon **(Tesauro, 1994)**
- Inventory Management **(Van Roy, Bertsekas, Lee, & Tsitsiklis, 1996)**
- Dynamic Channel Allocation **(e.g. Singh & Bertsekas, 1997)**
- Elevator Scheduling **(Crites & Barto, 1998)**
- Robocup Soccer **(e.g. Stone & Veloso, 1999)**
- Many Robots **(navigation, bi-pedal walking, grasping, switching between skills, ...)**
- Helicopter Control **(e.g. Ng, 2003, Abbeel & Ng, 2006)**
- More Applications **http://neuromancer.eecs.umich.edu/cgi-bin/twiki/view/Main/SuccessesOfRL**

Bayes RL

Value Function

- **State Value Function:**

$$V^{\mu}(x) = \mathbf{E}_{\mu}\left[\sum_{t=0}^{\infty} \gamma^t \bar{R}(x_t, \mu(x_t)) | x_0 = x\right]$$

- **State-Action Value Function:**

$$Q^{\mu}(x,a) = \mathbf{E}_{\mu}\left[\sum_{t=0}^{\infty} \gamma^t \bar{R}(x_t, a_t) | x_0 = x, a_0 = a\right]$$

Policy Evaluation

- **Finding the value function of a policy**

- **Bellman Equations**

$$V^{\mu}(x) = \sum_{a \in \mathcal{A}} \mu(a|x) \left[\bar{R}(x,a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x,a) V^{\mu}(x') \right]$$

$$Q^{\mu}(x,a) = \bar{R}(x,a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x,a) \sum_{a' \in \mathcal{A}} \mu(a'|x') Q^{\mu}(x',a')$$

Policy Optimization

- **Finding a policy μ^* maximizing $V^\mu(x) \quad \forall x \in \mathcal{X}$**

- **Bellman Optimality Equations**

$$V^*(x) = \max_{a \in \mathcal{A}} \left[\bar{R}(x,a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x,a) V^*(x') \right]$$

$$Q^*(x,a) = \bar{R}(x,a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x,a) \max_{a' \in \mathcal{A}} Q^*(x',a')$$

- **Note:** if $Q^*(x,a) = Q^{\mu^*}(x,a)$ is available, then an optimal action for state x is given by any $a^* \in \arg\max_a Q^*(x,a)$

POLICY OPTIMIZATION

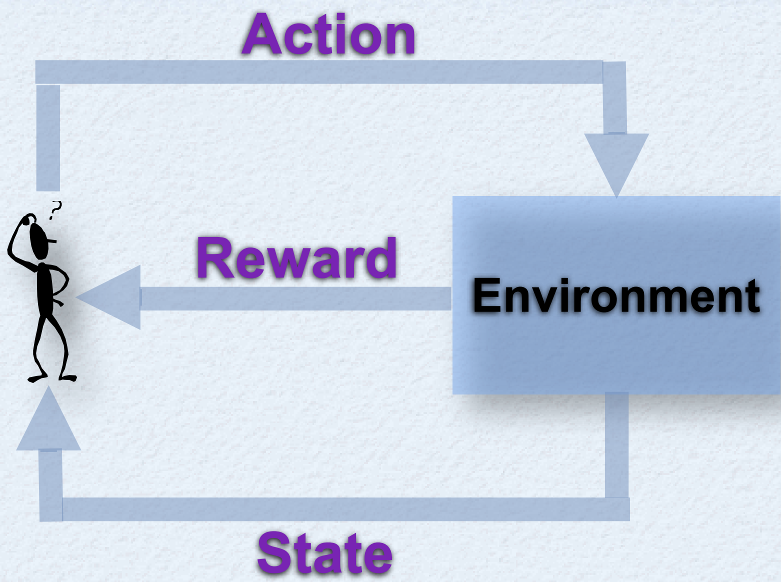
- **Value Iteration**

  - $V_0(x) = 0$

  - $$V_{t+1}(x) = \max_{a \in \mathcal{A}} \left[\bar{R}(x,a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x,a) V_t(x') \right]$$

system dynamics unknown

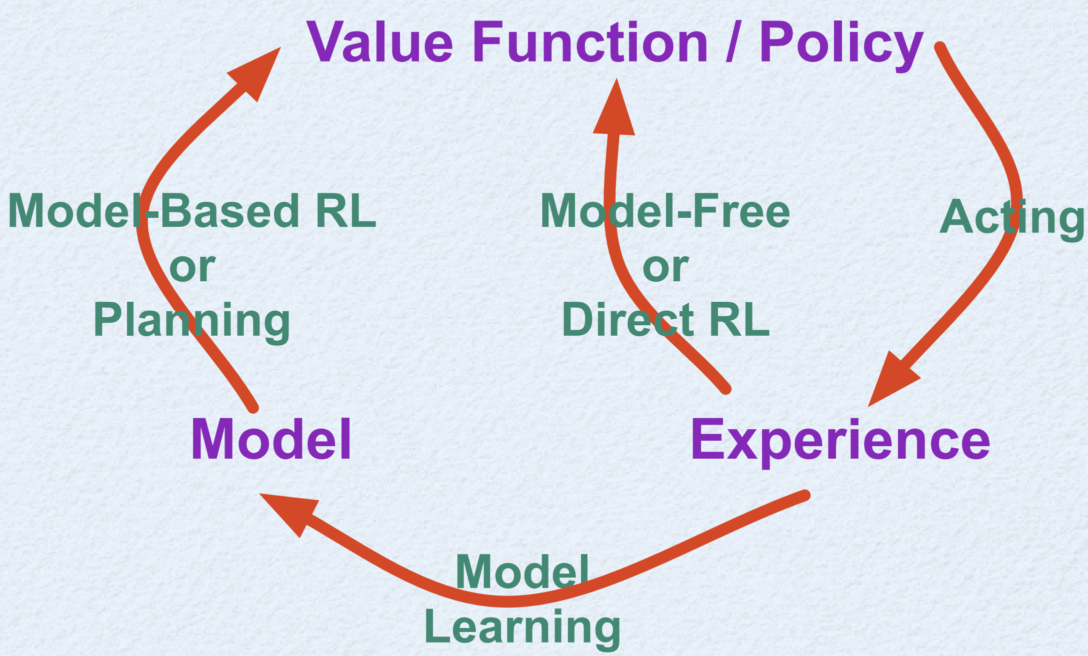
Reinforcement Learning (RL)



- **RL Problem:** Solve MDP when transition and/or reward models are unknown
- **Basic Idea:** use samples obtained from the agent's interaction with the environment to solve the MDP

Model-Based vs. Model-Free RL

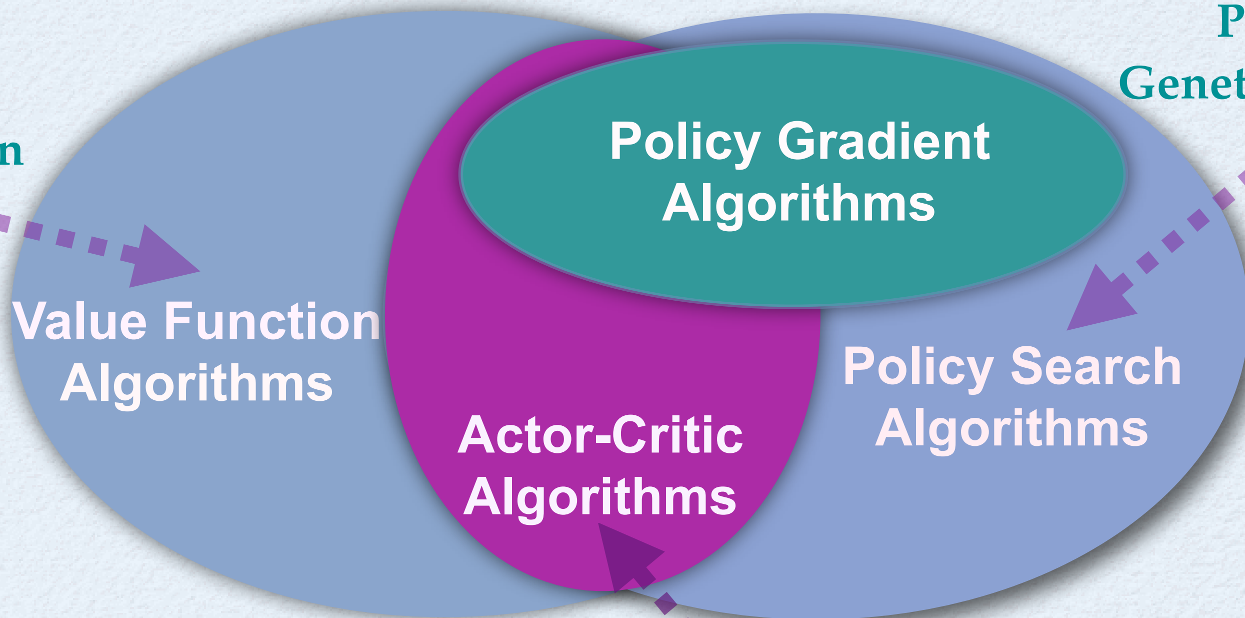
- **What is model?** state transition distribution and reward distribution
- **Model-Based RL:** model is not available, but it is explicitly learned
- **Model-Free RL:** model is not available and is not explicitly learned



Reinforcement Learning Solutions

SARSA
Q-learning
Value Iteration

PEGASUS
Genetic Algorithms



Sutton, et al. 2000
Konda & Tsitsiklis 2000
Peters, et al. 2005
Bhatnagar, Ghavamzadeh, Sutton 2007

Bayes RL

Learning Modes

- **Offline Learning**
  - Learning while interacting with a simulator

- **Online Learning**
  - Learning while interacting with the environment

Offline Learning

- Agent interacts with a simulator
- Rewards/costs do not matter
  - no exploration/exploitation tradeoff
- Computation time between actions is not critical
- Simulator can produce as much as data we wish
- **Main Challenge**
  - How to minimize time to converge to optimal policy

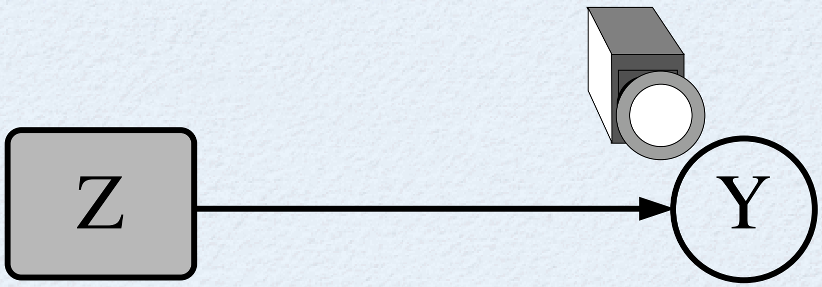
Bayes RL

Online Learning

- No simulator - Direct interaction with environment
- Agent receives reward/cost for each action
- **Main Challenges**
  - Exploration/exploitation tradeoff
    - Should actions be picked to maximize immediate reward or to maximize information gain to improve policy
  - Real-time execution of actions
  - Limited amount of data since interaction with environment is required

Bayesian Learning

The Bayesian Approach



- Z **- hidden process** , Y **- observable**
- **Goal:** infer Z from measurements of Y
- **Known:** $P(Y|Z)$ statistical dependence between Z and Y
- **Place prior over** Z **:** reflecting our uncertainty $P(Z)$
- **Observe:** $Y = y$
- **Compute posterior of** Z **:** $P(Z|Y=y) = \dfrac{P(y|Z)P(Z)}{\int P(y|Z')P(Z')dZ'}$

Bayesian Learning

- **Pros**
  - Principled treatment of uncertainty
  - Conceptually simple
  - Immune to overfitting **(prior serves as regularizer)**
  - Facilitates encoding of domain knowledge **(prior)**
- **Cons**
  - Mathematically and computationally complex
    - E.g. posterior may not have a closed form
  - How do we pick the prior?

Bayesian RL

+

- **Systematic method for inclusion and update of prior knowledge and domain assumptions**
  - Encode uncertainty about transition function, reward function, value function, policy, etc. with a probability distribution **(belief)**
  - Update belief based on evidence **(e.g., state, action, reward)**
- **Appropriately reconcile exploration with exploitation**
  - Select action based on belief
- **Providing full distribution, not just point estimates**
  - Measure of uncertainty for performance predictions **(e.g. value function, policy gradient)**

Bayesian RL

- **Model-based Bayesian RL**
  - Distribution over transition probability
- **Model-free Bayesian RL**
  - Distribution over value function, policy, or policy gradient
- **Bayesian inverse RL**
  - Distribution over reward
- **Bayesian multi-agent RL**
  - Distribution over other agents' policies