Announcements

Presentations in class next Tuesday and Thursday. Maximum of 10 minutes for your presentation. **Send me your slides no later than 9am the day of your presentation.**

Dec 5 Presentations

- Isaac, Adam, Shuzhe: Ventilation in Intensive Care Units
- Huiwen, Siyu: Treasure hunting
- Suyanpeng, Liyang: Pac Man
- Seok-Joo, Aditi, Ryan: Blackjack
- Andrew, Valerie, Anna: Fantasy Football Draft
- Derek, Chandra, Sajan: Robot Navigation

Last homework due next Tuesday

The movie AlphaGo is playing at the Michigan Theatre this Wed, 7pm

https://www.youtube.com/watch?v=8tq1C8spV_g

Model-Free Methods

Two major sources of challenges to solving MDPs are:

1) "curse of dimensionality"
2) "curse of modeling"

"Model-Free" methods are suited to problems of type 2, for which transition probabilities are not known

Model-Free Examples

Manufacturing

Elevator control

Health interventions

Robotics

Navigation

Robocup Soccer

Portfolio management

Power Systems

Protein Folding

Backgammon

Blackjack

Game of Go

Model-Free Methods

Model free approaches use **sample paths** to statistically estimate the value function under a particular policy

These methods are known under various names: *reinforcement learning, approximate dynamic programming, machine learning*

All that is necessary to use these methods is the ability to observe sample paths

Monte Carlo Sampling

Monte Carlo sampling is a common approach for estimating the expectation of functions of random variables

$$E_x[f(x)] \approx \frac{1}{N}\sum_{n=1}^{N} f(x^n)$$

Where x^n, $\mathrm{n} = 1, \ldots, N$ are random samples of random variable x.

Review: Sample Paths

A policy induces a particular stochastic process. In a **finite horizon MDP** the set of possible sample paths is

$$\Omega = S \times A \times S \times A \times \cdots \times A \times S = \{S \times A\}^{N-1} \times S$$

A **sample path** defines the states and actions for a realization of the induced stochastic process

$$(s_0, a_0, s_1, a_1, s_2, a_2, \ldots, \mathrm{a_{N-1}}, \mathrm{s_N})$$

Sampling a Markov Chain

Choose a probability distribution, q, for the initial state of the Markov chain with M states

$$q(s_1) + q(s_2) + \cdots + q(s_M) = 1$$

Randomly sample from q to select an initial state of the system, s_1.

At each subsequent stage t, choose an action according to policy, π_t, and randomly sample the next state using the transition probability matrix, $P_t(a_t)$.

Record the rewards at each stage:

$$r_1(s_1, \pi_1(s_1)), r_2(s_2, \pi_2(s_2)), \ldots., r_N(s_N, \pi_N(s_N))$$

Monte Carlo Policy Evaluation

Policies can be evaluated by sampling rewards to estimate the value function for a specific policy, π

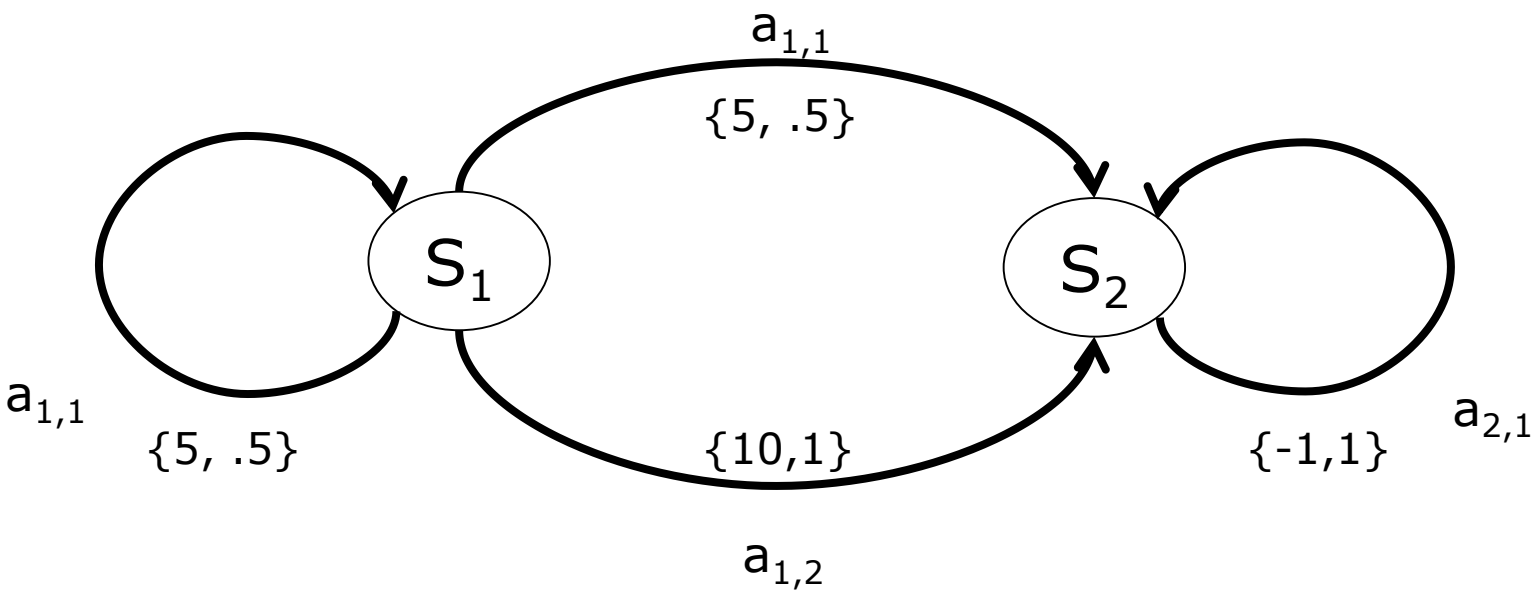
$$\tilde{v}^{\pi}(s_0) = \sum_{t=0}^{N-1} \lambda^t r_t(s_t, a_t) + \lambda^n r_N(s_N)$$

As N $\rightarrow \infty$ $\tilde{v}^{\pi}(s_0) \rightarrow v^{\pi}(s_0)$.

In practice the number of sample patha, N, must be chosen to tradeoff between (a) some desired level of confidence and (b) a computational budget.

Example Revisited: 2 State MDP

In state S_1 actions $a_{1,1}$ and $a_{1,2}$ are available; in state S_2 only $a_{2,1}$ is available. Rewards and transition probabilities are defined below as {r,p}



Exercise: Assume you start in state s_1, i.e., $q(s_1) = 1, q(s_2) = 0$. Sample the Markov chain to evaluate each policy to find the best policy. Assume λ = 0.5.

Monte Carlo Policy Evaluation

MC Policy evaluation can be done regardless of whether or not you know the initial distribution, q, or the transition probability matrix, P.

You only need to be able to observe the states, actions, and rewards

$$s_1, a_1, r_1(s_1, a_1) \rightarrow s_2, a_2, r_2(s_2, a_2) \rightarrow s_3, a_3, r_3(s_3, a_3) \rightarrow \cdots$$

This is what is meant by **"model free"**

Monte Carlo Policy Iteration

MC Policy Iteration parallels standard policy iteration:

1) Select an initial policy to evaluate
2) Evaluate the current policy using MC Policy Evaluation
3) Improve the current policy
4) Stop if convergence criteria is satisfied; Otherwise go to Step 2.

This is an example of an **off-policy** method

Example: multi-armed bandit

Following is an example of a situation in which you must "**learn as you go**"

Consider a betting game in which your friend holds two coins: 1 coin is fair, the other is biased towards landing heads up.

You only know your friend holds two different types of coins but you don't know the likelihood of each turning up a head.

Each turn you get to select the coin your friend will flip. If you win you get $1 if you lose you lose $1.

How would you play this game?

Applications: gambling, medical treatment decisions, financial portfolio management, internet advertising

Example: multi-armed bandit

The action is which "arm", a, to try at each decision epoch, and the expected reward for this action is $Q_t(a)$.

Since $Q_t(a)$ is not known exactly. It must be estimated as:

$$\widetilde{Q_t}(a) = \frac{r_1 + r_2 + \cdots + r_{k_a}}{k_a}$$

Where k_a is the number of times arm a has been pulled.

As $k_a \to \infty$ $\widetilde{Q_t}(a) \to Q_t(a)$, thus sampling each arm an infinite number of times will identify the optimal action

$$a^* = argmax_{a \in A}\{Q_t(a)\}.$$

Example: multi-armed bandit

The simplest learning-based policy is the greedy policy that selects the best action using the approximate expected reward for each action:

$$\tilde{a} = argmax\{\widetilde{Q_t}(a)\}$$

This approach often performs poorly because you do not adequately **explore** the different potential actions.

Example: multi-armed bandit

The $\epsilon - greedy$ method **explores** the action set using a randomized policy.

At each step the $\epsilon - greedy$ method chooses one of two options:

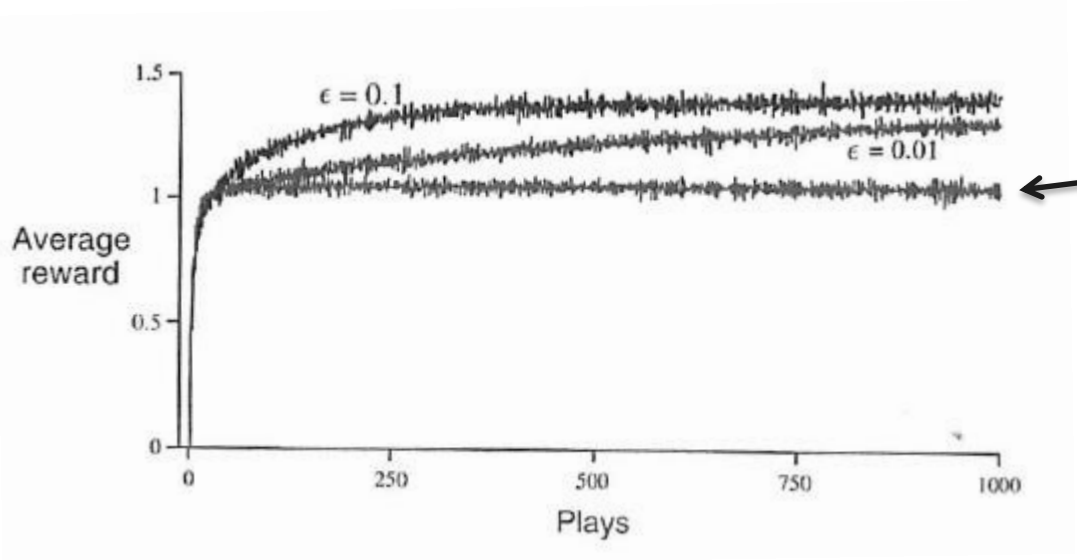
1) **Exploitation**: with probability $1 - \epsilon$ select the action the maximizes rewards based on the current estimate of $Q_t(a)$

2) **Exploration**: with probability ϵ randomly select an action

As $\mathrm{k_a} \to \infty, Q_t(a) \to Q_t^*(a)$, and the optimal action is selected with probability at least $1 - \epsilon$.

Example: multi-armed bandit

The following results are averages across 2000 randomly generated 10-armed bandit problems for 3 different greedy methods.



Taken from "Reinforcement Learning: An Introduction", By Sutton and Barto, MIT Press

Learn More

Abhijit Gosavi, 2009, Reinforcement Learning: A Tutorial Survey and Recent Advances, INFORMS Journal on Computing, 212, 178-192. (on Canvas)

“Reinforcement Learning: An Introduction”, By Sutton and Barto, MIT Press