Markov Decision Processes (MDPs)

Industrial AI Lab.

Prof. Seungchul Lee

Today

- Markov Chain
- Markov Reward Process
- Markov Decision Process

Markov Chain

Sequential Processes

- Most classifiers ignored the sequential aspects of data

- Consider a system which can occupy one of *N* discrete states or categories

$$q_t \in \{S_1, S_2, \ldots, S_N\}$$

- We are interested in stochastic systems, in which state evolution is random
- Any joint distribution can be factored into a series of conditional distributions

$$p(q_0, q_1, \ldots, q_T) = p(q_0)p(q_1|q_0)p(q_2|q_1q_0)\cdots$$

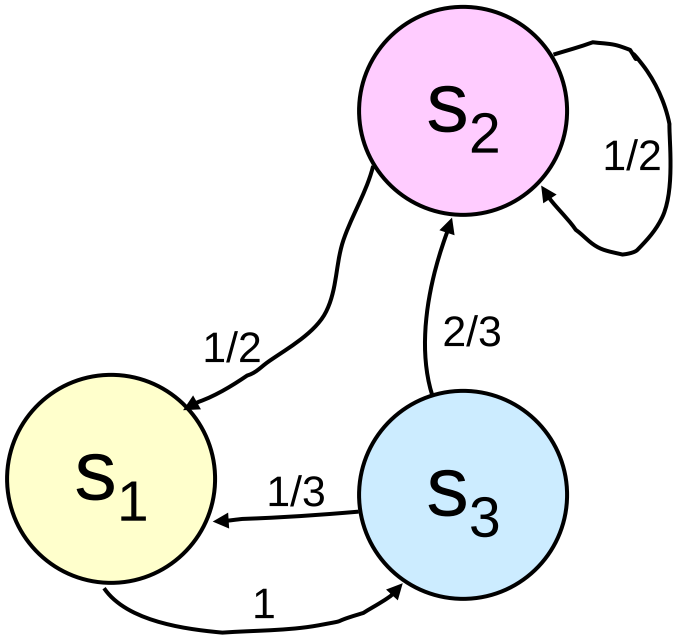
Almost impossible to compute !

Markov Process

$$p(q_0, q_1, \dots, q_T) = p(q_0)p(q_1|q_0)p(q_2|q_1q_0)p(q_3|q_2q_1q_0)\cdots$$

$$p(q_0, q_1, \dots, q_T) = p(q_0)p(q_1|q_0)p(q_2|q_1)p(q_3|q_2)\cdots$$

Possible and tractable



Markov Process

- (Assumption) for a Markov process, the next state depends only on the current state:

$$p(q_{t+1}|q_t, \cdots, q_0) = p(q_{t+1}|q_t)$$

- More clearly

$$P\left(q_{t+1} = s_j | q_t = s_i\right) = P\left(q_{t+1} = s_j | q_t = s_i \text{, any earlier history}\right)$$

- Given current state, the past does not matter
- The state captures all relevant information from the history
- The state is a sufficient statistic of the future

State Transition Matrix

- For a Markov state s and successor state s', the state transition probability is defined by

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$$

- State transition matrix P defines transition probabilities from all states s to all successor states s',

$$\mathcal{P} = \begin{array}{cc} & \textit{to} \\ \textit{from} & \begin{bmatrix} \mathcal{P}_{11} & \ldots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \ldots & \mathcal{P}_{nn} \end{bmatrix} \end{array}$$

POSTECH

Markov Process

- A Markov process is a memoryless random process, i.e., a sequence of random states $s_1, s_2, \cdots$ with the Markov property

Definition

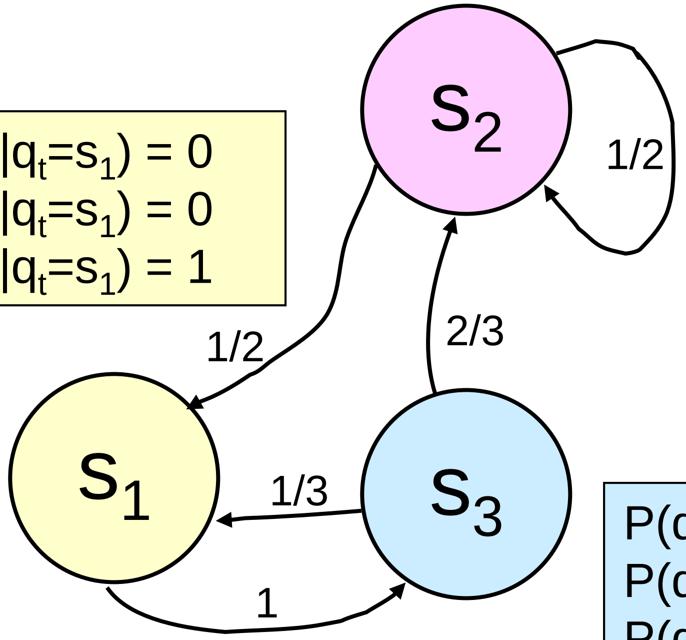
A *Markov Process* (or *Markov Chain*) is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$

State Transition Matrix

$P(q_{t+1}=s_1|q_t=s_2) = 1/2$
$P(q_{t+1}=s_2|q_t=s_2) = 1/2$
$P(q_{t+1}=s_3|q_t=s_2) = 0$

$P(q_{t+1}=s_1|q_t=s_1) = 0$
$P(q_{t+1}=s_2|q_t=s_1) = 0$
$P(q_{t+1}=s_3|q_t=s_1) = 1$



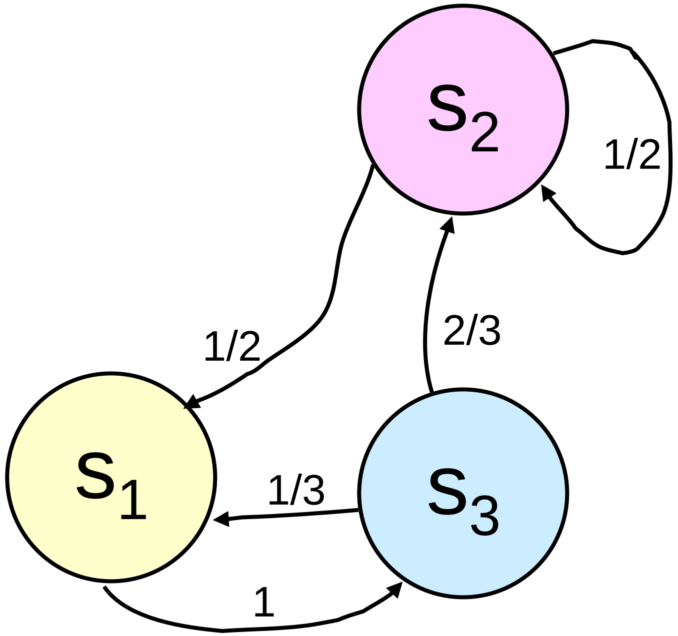
$P(q_{t+1}=s_1|q_t=s_3) = 1/3$
$P(q_{t+1}=s_2|q_t=s_3) = 2/3$
$P(q_{t+1}=s_3|q_t=s_3) = 0$

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \\ 1/3 & 2/3 & 0 \end{bmatrix}$$

Property of *P* Matrix

- Sum of the elements on each row yields 1

$$\sum_{j \in S} p_{i,j} = 1$$

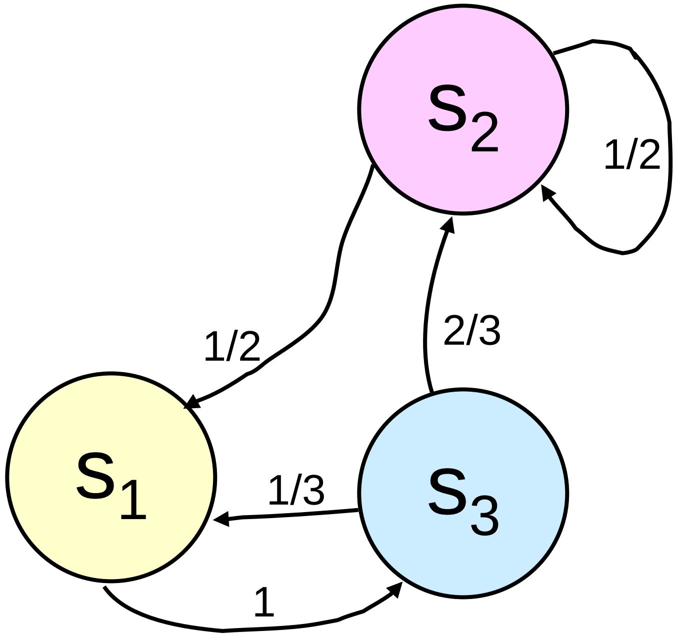


$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \\ 1/3 & 2/3 & 0 \end{bmatrix}$$

Property of P Matrix

- Sum of the elements on each row yields 1

$$\sum_{j \in S} p_{i,j} = 1$$

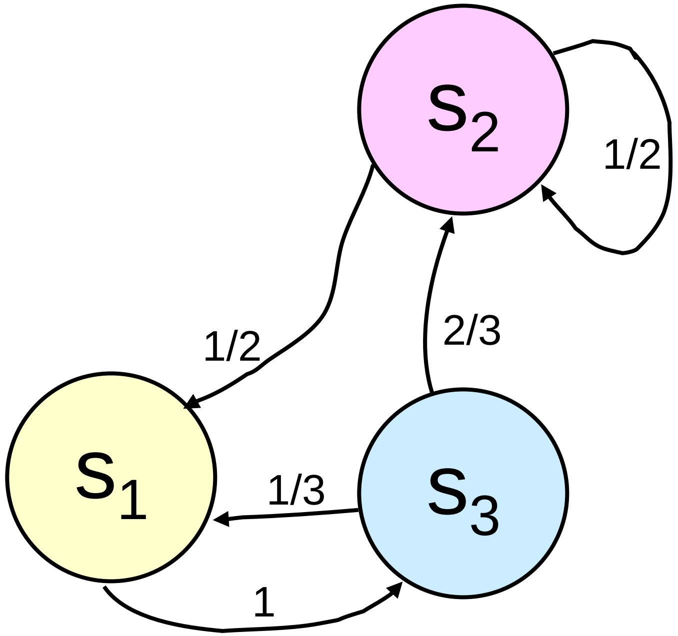


$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \\ 1/3 & 2/3 & 0 \end{bmatrix}$$

- Question: P^2 and P^n (will discuss later)

Markov Chain Components

1. a finite set of *N* states, S = { $S_1, \cdots, S_N$ }
2. a state transition probability, $\mathrm{P}=\{a_{ij}\}_{M\times M}$, $1 \leq i,j \leq M$
3. an initial state probability distribution, $\pi = \{\pi_i\}$

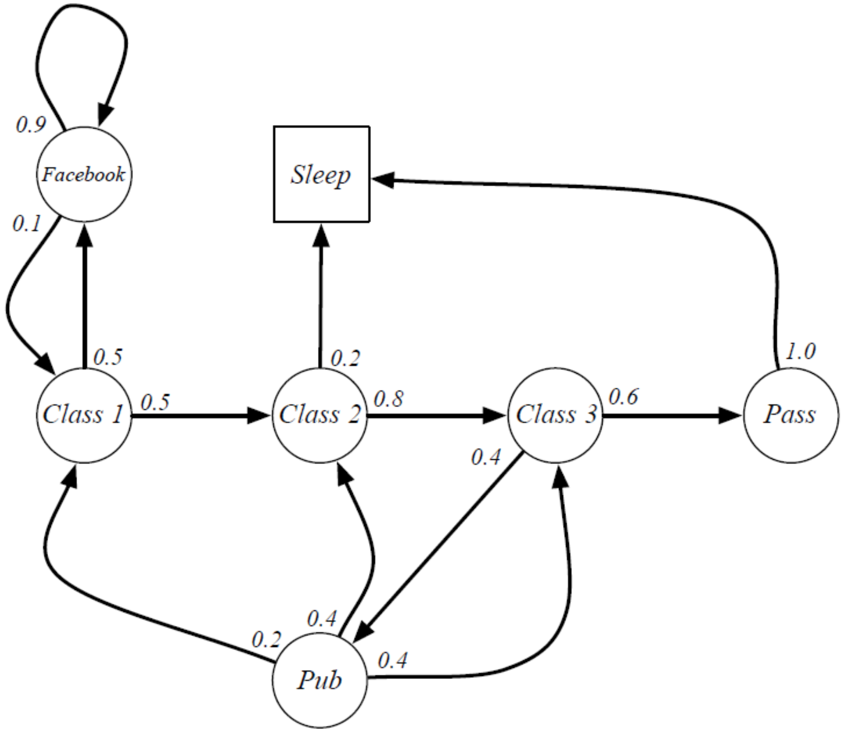


$$P=\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \\ 1/3 & 2/3 & 0 \end{bmatrix}$$

- Passive stochastic behavior

Student Markov Chain Episodes

- Starting from S_1 = Class 1



- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

$$
\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array}
\begin{array}{c}
\begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\
\left[\begin{array}{ccccccc}
 & 0.5 & & & & 0.5 & \\
 & & 0.8 & & & & 0.2 \\
 & & & 0.6 & 0.4 & & \\
 & & & & & & 1.0 \\
0.2 & 0.4 & 0.4 & & & & \\
0.1 & & & & & 0.9 & \\
 & & & & & & 1
\end{array}\right]
\end{array}
$$

Chapman-Kolmogorov Equation

- (1-step transition probabilities) For a Markov chain on a finite state space, S = { $S_1, \cdots, S_N$}, with transition probability matrix P and initial distribution $\pi = \left\{\pi_i^{(0)}\right\}$ (row vector) then the distribution of $X(1)$ is given by

$$\begin{bmatrix} \pi_1^{(1)} & \pi_2^{(1)} & \pi_3^{(1)} \end{bmatrix} = \begin{bmatrix} \pi_1^{(0)} & \pi_2^{(0)} & \pi_3^{(0)} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

Chapman-Kolmogorov Equation

- (2-step transition probabilities) For a Markov chain on a finite state space, S = { $S_1, \cdots, S_N$}, with transition probability matrix P and initial distribution $\pi = \left\{\pi_i^{(0)}\right\}$ (row vector) then the distribution of $X(2)$ is given by

$$\begin{bmatrix}\pi_1^{(2)} & \pi_2^{(2)} & \pi_3^{(2)}\end{bmatrix} = \begin{bmatrix}\pi_1^{(1)} & \pi_2^{(1)} & \pi_3^{(1)}\end{bmatrix} P = \begin{bmatrix}\pi_1^{(0)} & \pi_2^{(0)} & \pi_3^{(0)}\end{bmatrix} P^2$$

Chapman-Kolmogorov Equation

- (n-step transition probabilities) For a Markov chain on a finite state space, S = { $S_1, \cdots, S_N$}, with transition probability matrix P and initial distribution $\pi = \left\{\pi_i^{(0)}\right\}$ (row vector) then the distribution of $X(n)$ is given by

$$\begin{bmatrix} \pi_1^{(n)} & \pi_2^{(n)} & \pi_3^{(n)} \end{bmatrix} = \begin{bmatrix} \pi_1^{(n-1)} & \pi_2^{(n-1)} & \pi_3^{(n-1)} \end{bmatrix} P = \begin{bmatrix} \pi_1^{(0)} & \pi_2^{(0)} & \pi_3^{(0)} \end{bmatrix} P^n$$

- P^n: n-step transition probabilities

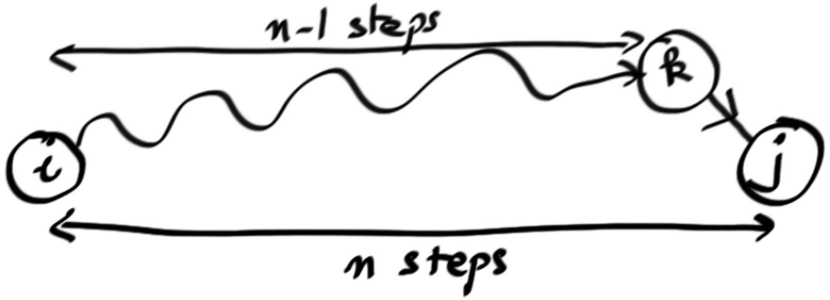
n-step Transition Probability

- $p_{ij}(n) = P[X_n = j | X_0 = i]$
- $p_{ij} = p_{ij}(1) = P[X_1 = j | X_0 = i]$

- Key recursion:

$$p_{ij}(n) = \sum_{k=1}^{N} p_{ik}(n-1)\, p_{kj}(1)$$

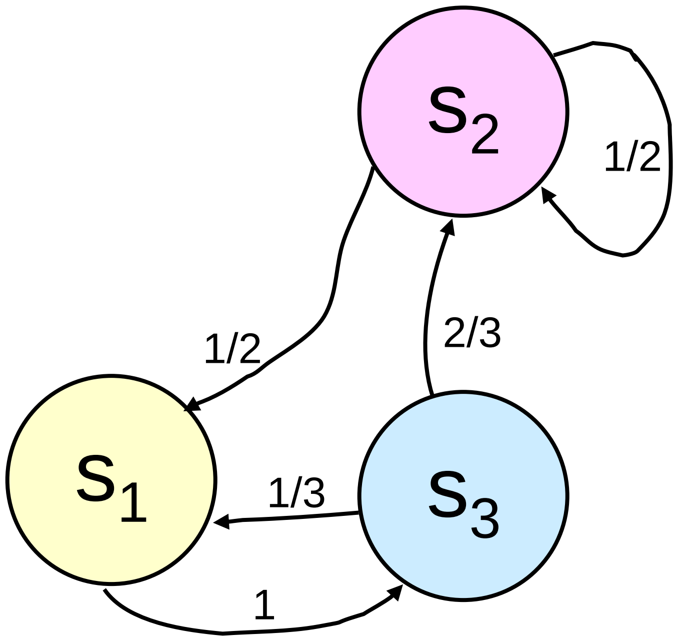
$$i \rightarrow k \text{ and } k \rightarrow j \text{ imply } i \rightarrow j$$



Example

	n = 1	n = 2	n = 3
$p_{11}(n)$			
$p_{12}(n)$			
$p_{13}(n)$			

$$\begin{bmatrix}\pi_1 & \pi_2 & \pi_3\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}$$



Stationary Distribution

- Steady-state behavior
- Does $p_{ij}(n) = P[X_n = j | X_0 = i]$ converge to some π_j?

- Take the limit as $n \to \infty$

$$p_{ij}(n) = \sum_{k=1}^{N} p_{ik}(n-1)\, p_{kj}$$

$$\pi_j = \sum_{k=1}^{N} \pi_k\, p_{kj}$$

- Need also $\sum_j \pi_j = 1$

$$\boxed{\pi = \pi P}$$

- How to compute
  - Eigen-analysis
  - Fixed-point iteration

Markov Reward Process

Markov Chains with Rewards

- Suppose that each transition in a Markov chain is associated with a reward, r
- As the Markov chain proceeds from state to state, there is an associated sequence of rewards
- Discount factor γ

- Later, we will study dynamic programming and Markov decision theory $\Rightarrow$ Markov Decision Process (MDP)
  - These topics include a *decision maker*, *policy maker*, or *control* that modify both the transition probabilities and the rewards at each trial of the Markov chain.

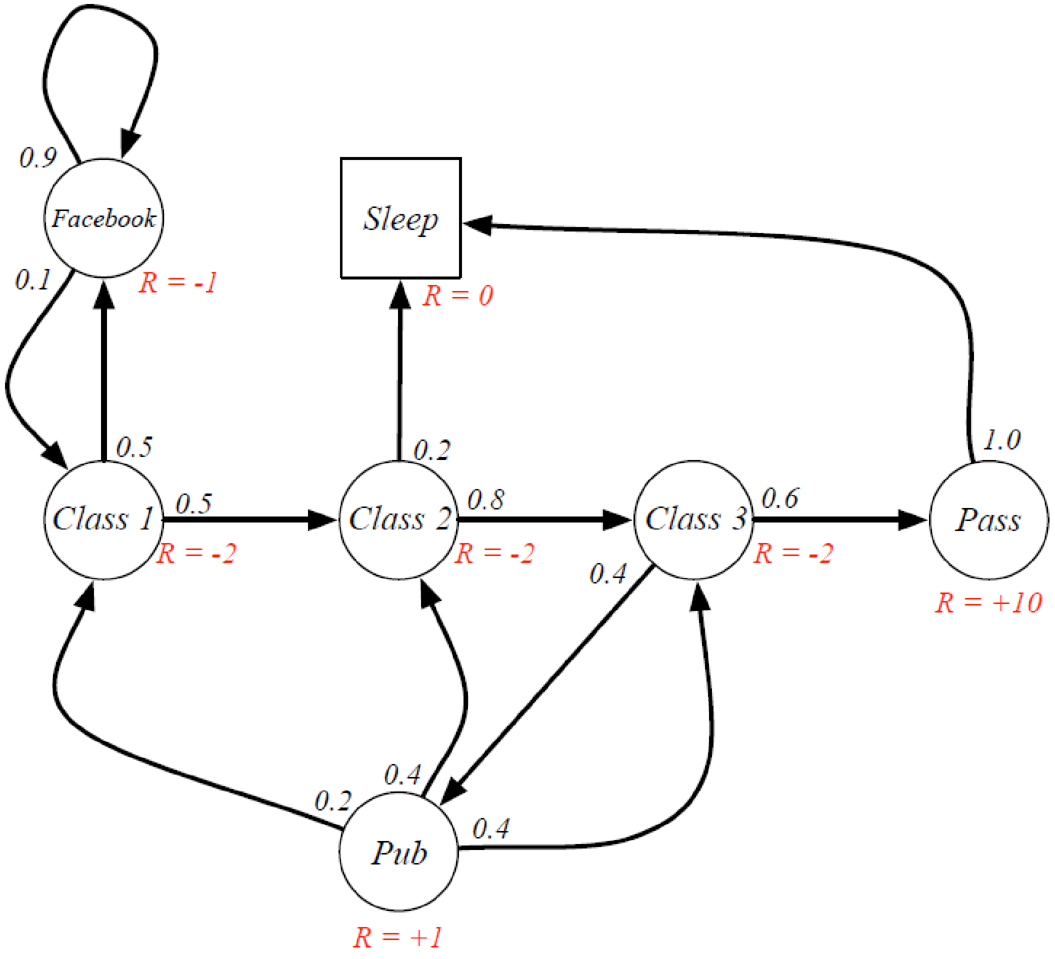
Markov Reward Process (MRP)

Definition

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}\left[R_{t+1} \mid S_t = s\right]$
- γ is a discount factor, $\gamma \in [0, 1]$

Student MRP


0.9
Facebook
0.1
R = -1
Sleep
R = 0
0.5
0.2
1.0
Class 1
0.5
R = -2
Class 2
0.8
R = -2
Class 3
0.6
R = -2
Pass
R = +10
0.4
0.4
0.2
0.4
Pub
R = +1

Reward over Multiple Transitions

- Return

Definition

The *return* G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Value Function

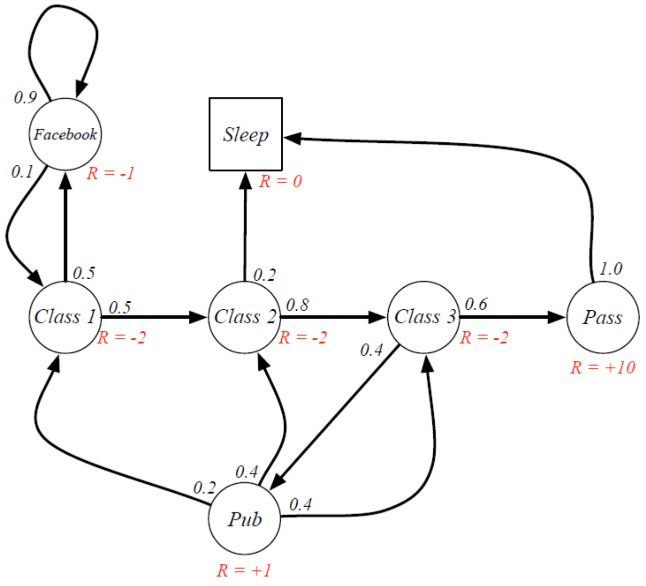
- The value function $v(s)$ gives the long-term value of state s

Definition

The *state value function* $v(s)$ of an MRP is the expected return starting from state s

$$v(s) = \mathbb{E}\left[G_t \mid S_t = s\right]$$

Student MRP Returns

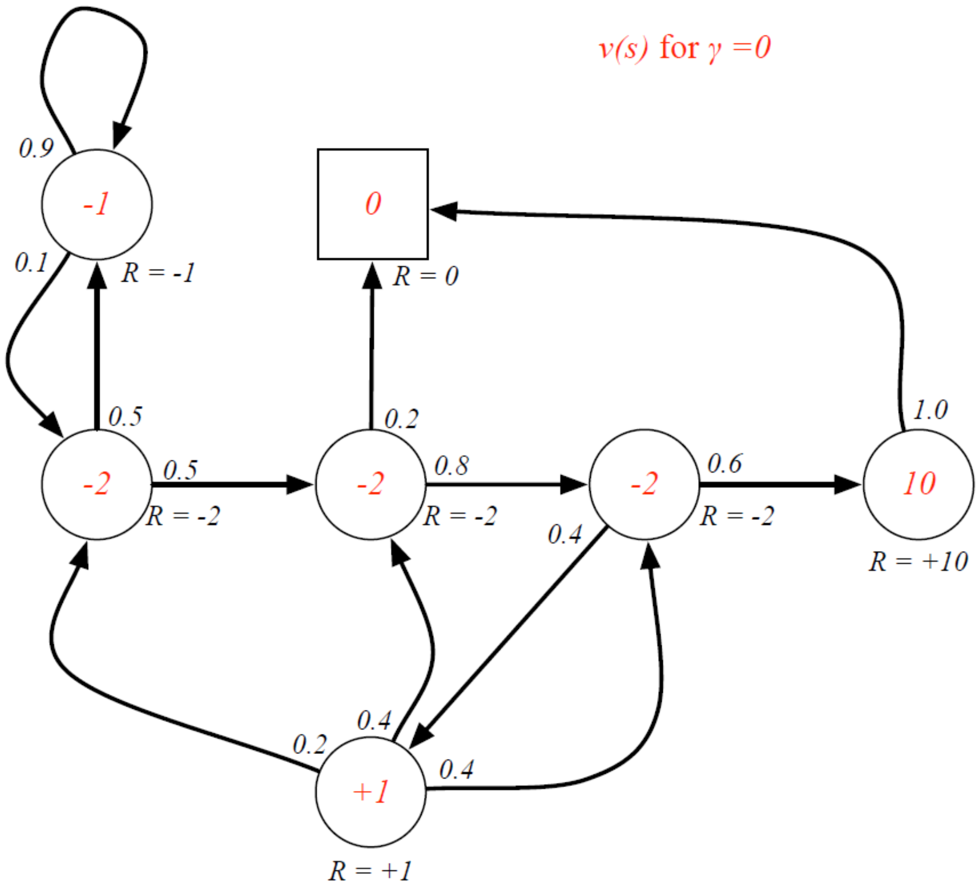


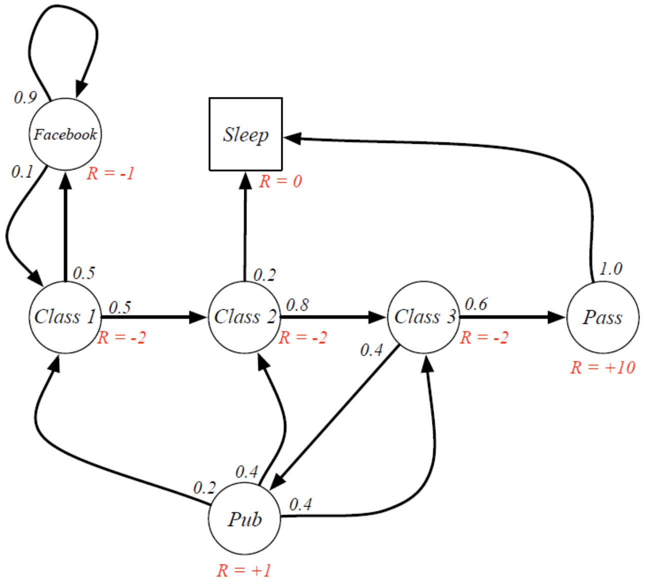
Sample returns for Student MRP:
Starting from $S_1 = \text{C1}$ with $\gamma = \frac{1}{2}$

$$G_1 = R_2 + \gamma R_3 + \ldots + \gamma^{T-2} R_T$$

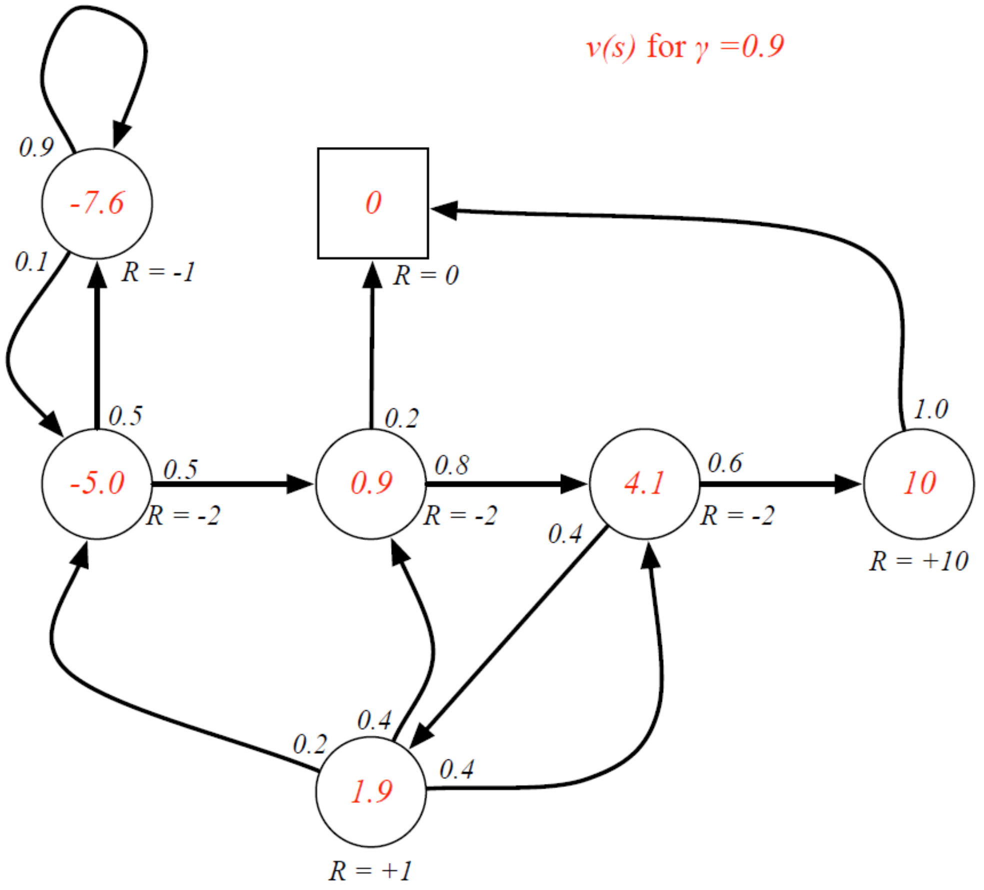
C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8}$	$=$	-2.25
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16}$	$=$	-3.125
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	$=$	-3.41
C1 FB FB C1 C2 C3 Pub C1 ...	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	$=$	-3.20
FB FB FB C1 C2 C3 Pub C2 Sleep			

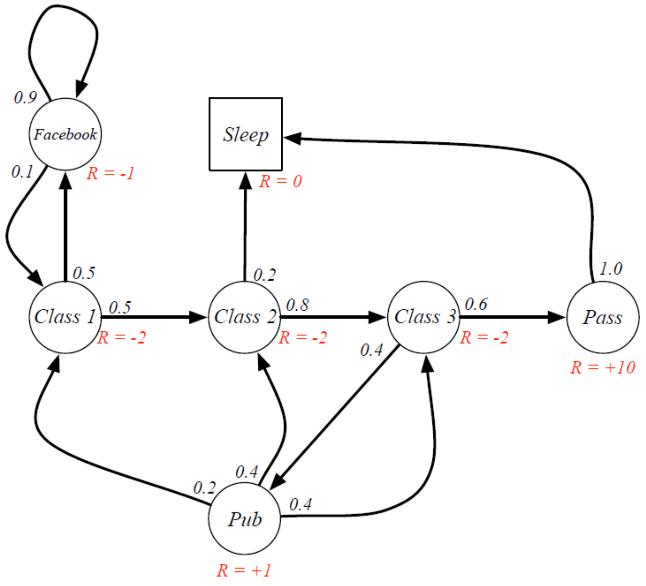
State-Value Function for Student MRP (1)



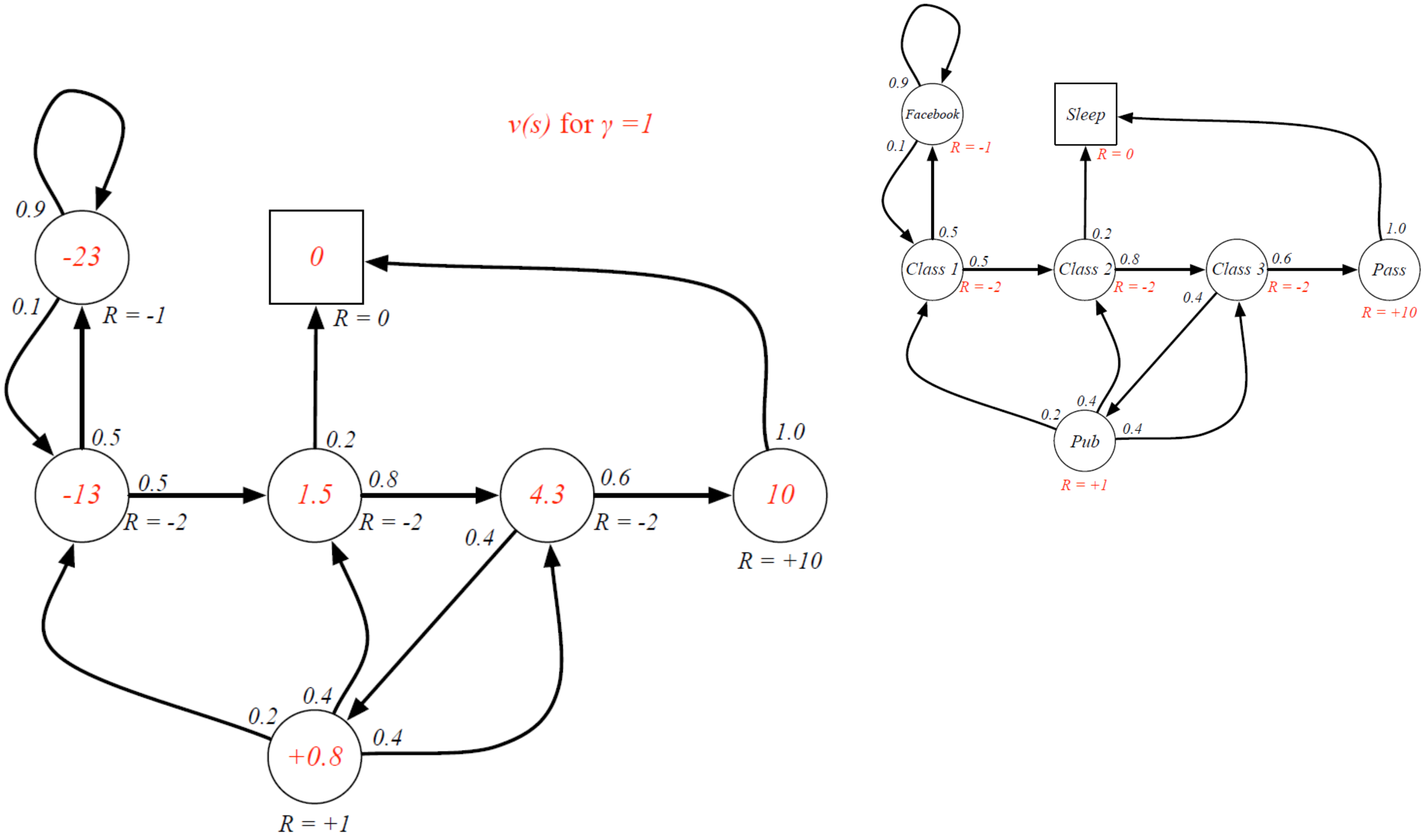


State-Value Function for Student MRP (2)





State-Value Function for Student MRP (3)



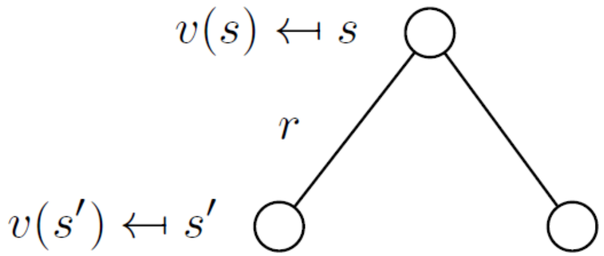
Bellman Equations for MRP (1)

- The value function can be decomposed into two parts:
  - Immediate reward R_{t+1}
  - Discounted value of successor state $\gamma v(S_{t+1})$

$$
\begin{aligned}
v(s) &= \mathbb{E}\left[G_t \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma\left(R_{t+2} + \gamma R_{t+3} + \ldots\right) \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]
\end{aligned}
$$

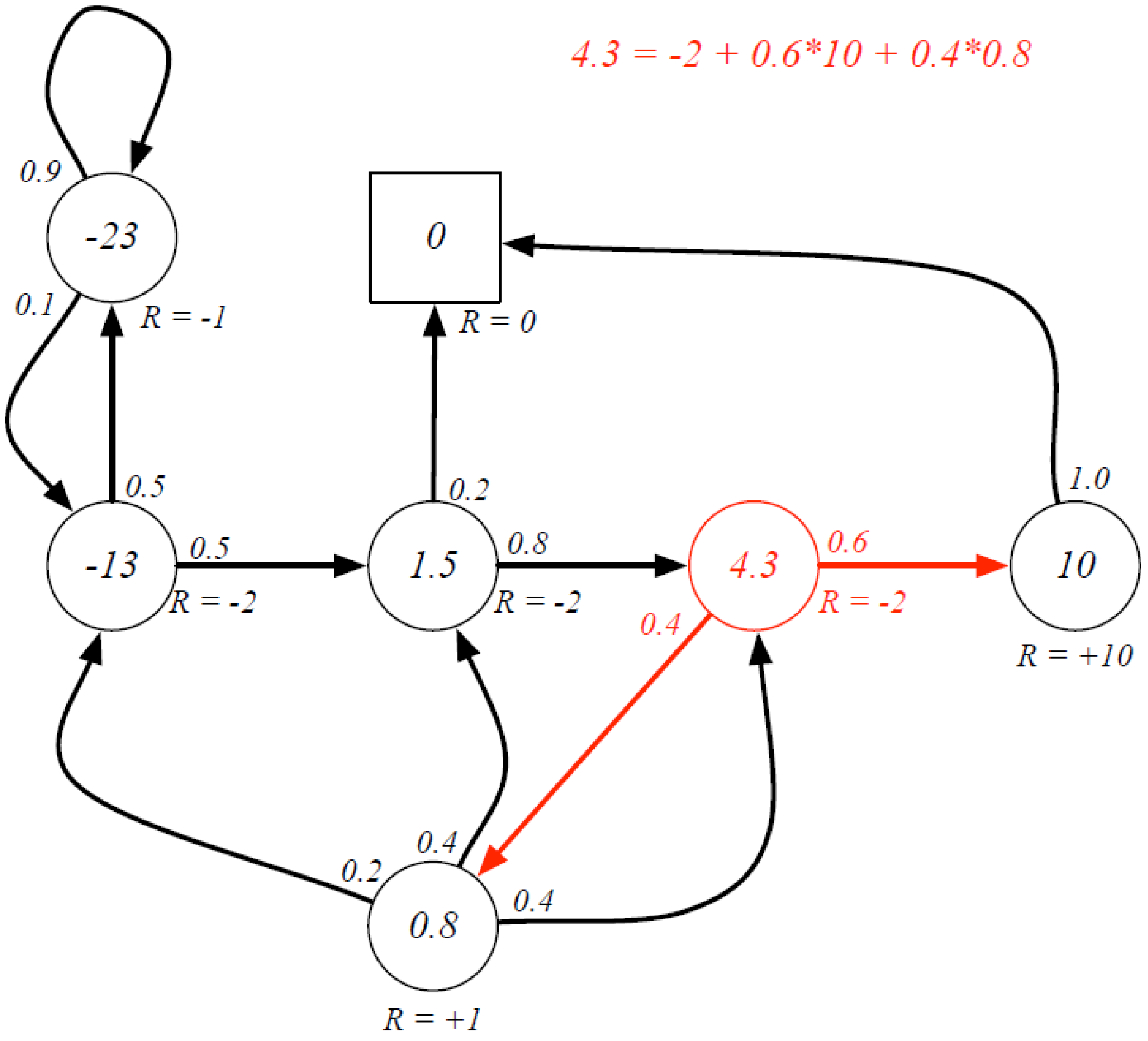
Bellman Equations for MRP (2)

$$v(s) = \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]$$



$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

Bellman Equation for Student MRP



Bellman Equation in Matrix Form

- The Bellman equation can be expressed concisely using matrices,

$$v = \mathcal{R} + \gamma \mathcal{P} v$$

where v is a column vector with one entry per state

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_n \end{bmatrix} + \gamma \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ & \vdots & \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

Solving the Bellman Equation

- The Bellman equation is a linear equation
- It can be solved directly:

$$\begin{aligned} v &= \mathcal{R} + \gamma \mathcal{P} v \\ (I - \gamma \mathcal{P})\, v &= \mathcal{R} \\ v &= (I - \gamma \mathcal{P})^{-1} \mathcal{R} \end{aligned}$$

- Direct solution only possible for small MRP
- There are many *iterative* methods for large MRP
  - Dynamic programming
  - Monte-Carlo simulation
  - Temporal-difference learning

Quiz

- A miner is trapped in a mine containing three doors.
  - The first door leads to a tunnel that takes him to safety after two hours of travel.
  - The second door leads to a tunnel that returns him to the mine after three hours of travel.
  - The third door leads to a tunnel that returns him to his mine after five hours.
- Assuming that the miner is at all times equally likely to choose any one of the doors, what is the expected length of time until the miner reaches safety?

Markov Decision Process

Markov Decision Process

- So far, we analyzed the passive behavior of a Markov chain with rewards

- A Markov decision process (MDP) is a Markov reward process with decisions (or actions).

Definition

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

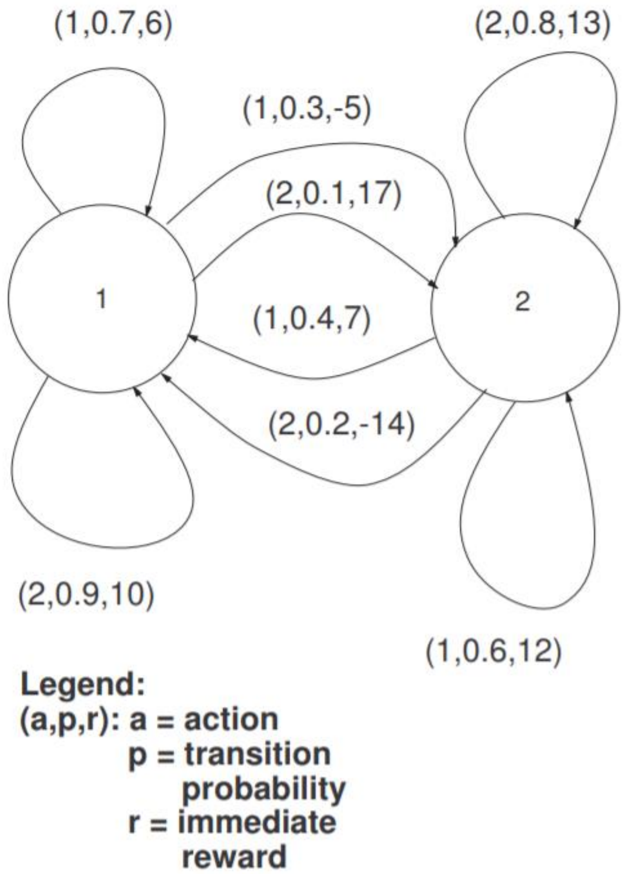
- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'}^{a} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$
- γ is a discount factor $\gamma \in [0, 1]$.

Example

- P_a: transition probability matrix for action a
- R_a: transition reward matrix for action a

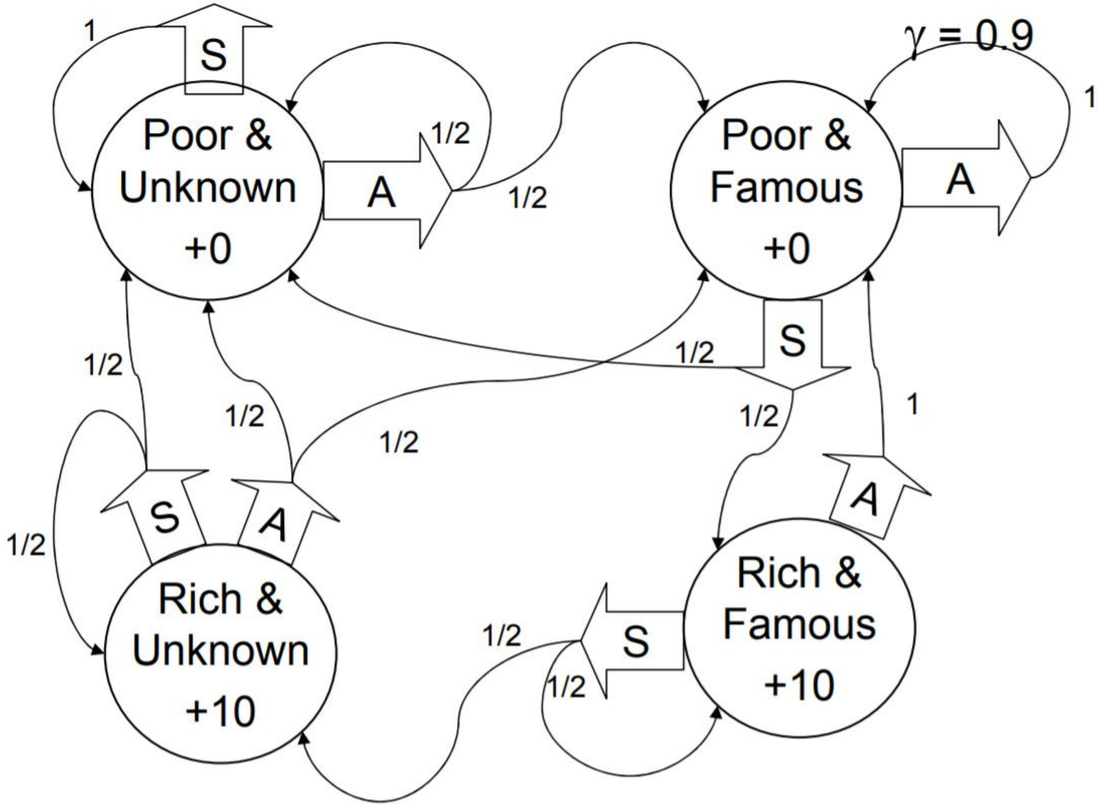
$$\mathbf{P}_1 = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}; \mathbf{P}_2 = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix};$$

$$\mathbf{R}_1 = \begin{bmatrix} 6 & -5 \\ 7 & 12 \end{bmatrix}; \mathbf{R}_2 = \begin{bmatrix} 10 & 17 \\ -14 & 13 \end{bmatrix}.$$



Example

- You run a startup company.
  - In every state, you must choose between Saving money or Advertising

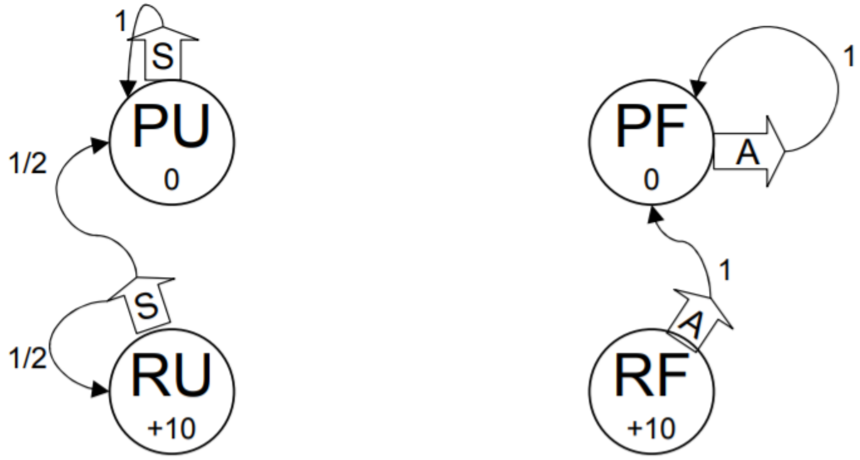


Policy

- A policy is a mapping from states to actions, $\pi: S \rightarrow A$
- Example: two policies

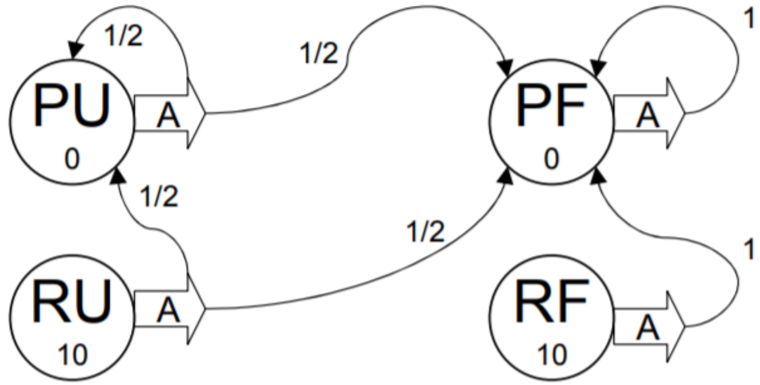
Policy Number 1:

STATE →	ACTION
PU	S
PF	A
RU	S
RF	A



Policy Number 2:

STATE →	ACTION
PU	A
PF	A
RU	A
RF	A



Policies

- A policy is a mapping from states to actions, $\pi: S \rightarrow A$

- A policy fully defines the behavior of an agent

- Let P^{π} be a matrix containing probabilities for each transition under policy π

- Given an MDP $\mathcal{M} = \langle S, A, P, R, \gamma \rangle$ and a policy π
  - The state sequence $s_1, s_2, \cdots$ is a Markov process $\langle S, P^{\pi} \rangle$
  - The state and reward sequence is a Markov reward process $\langle S, P^{\pi}, R^{\pi}, \gamma \rangle$

Questions on MDP Policy

- How many possible policies in our example?
- Which of the above two policies is best?
- How do you compute the *optimal* policy?

State-Value Function

Definition

The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$v_\pi(s) = \mathbb{E}_\pi\left[G_t \mid S_t = s\right]$$

- Given the policy π, the state-value function can again be decomposed into immediate reward plus discounted value of successor state (recursively)

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right]$$

Bellman Expectation Equation

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right]$$

$$\Downarrow$$

$$v_\pi(s) = R(s) + \gamma \sum_{s' \in S} P^\pi_{ss'}\, v_\pi\left(s'\right)$$

- The Bellman expectation equation can be expressed concisely in a matrix form,

$$v_\pi = R + \gamma P^\pi v_\pi$$

with direct solution

$$v_\pi = (I - \gamma P^\pi)^{-1} R$$

Optimal Policy and Optimal Value Function

- The optimal policy is the policy that achieves the highest value for every state

$$\pi^*(s) = \arg\max_{\pi} v_\pi(s)$$

  and its optimal value function

- We can directly define the *optimal value function* using Bellman optimality equation

$$v^*(s) = R(s) + \gamma \max_a \sum_{s' \in S} P^a_{ss'} \, v^*(s')$$

  and *optimal policy* is simply the action that attains this max

$$\pi^*(s) = \arg\max_a \sum_{s' \in S} P^a_{ss'} \, v_\pi(s')$$

Computing the Optimal Policy

- Value iteration
  - According to Bellman optimality equation

1) initialize an estimate for the value function arbitrarily

$$v(s) \leftarrow 0 \quad \forall s \in S$$

2) Repeat, update

$$v(s) \leftarrow R(s) + \gamma \max_a \sum_{s' \in S} P_{ss'}^a \, v\left(s'\right), \quad \forall s \in S$$

Solving the Bellman Optimality Equation

- Bellman Optimality Equation is non-linear
- No closed form solution (in general)

- (Will learn later) many iterative solution methods
  - Value Iteration
  - Policy Iteration
  - Q-learning
  - SARSA

➢ You will get into details in the course of reinforcement learning