Markov Reward Process (MRP)

Prof. Seungchul Lee

Industrial AI Lab.

Source

- David Silver's Lecture (DeepMind)
  - UCL homepage for slides (http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html)
  - DeepMind for RL videos (https://www.youtube.com/watch?v=2pWv7GOvuf0)
  - An Introduction to Reinforcement Learning, Sutton and Barto pdf

- CMU by Zico Kolter
  - http://www.cs.cmu.edu/~zkolter/course/15-780-s14/lectures.html
  - https://www.youtube.com/watch?v=un-FhSC0HfY&hd=1

- Deep RL Bootcamp by Rocky Duan
  - https://sites.google.com/view/deep-rl-bootcamp/home
  - https://www.youtube.com/watch?v=qO-HUo0LsO4

- Stanford Univ. by Serena Yeung
  - https://www.youtube.com/watch?v=lvoHnicueoE&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=15&t=1337s

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 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 $\langle S, P, R, \gamma \rangle$

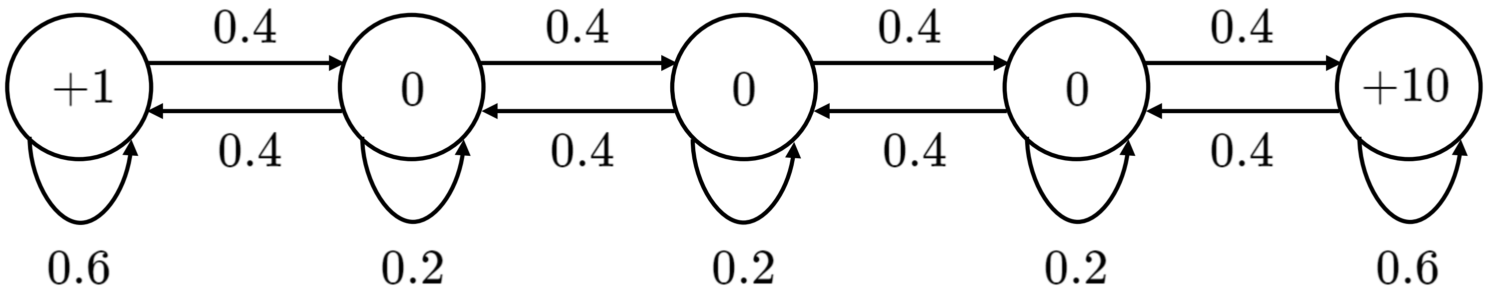
- S is a finite set of states
- P is a state transition probability matrix

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

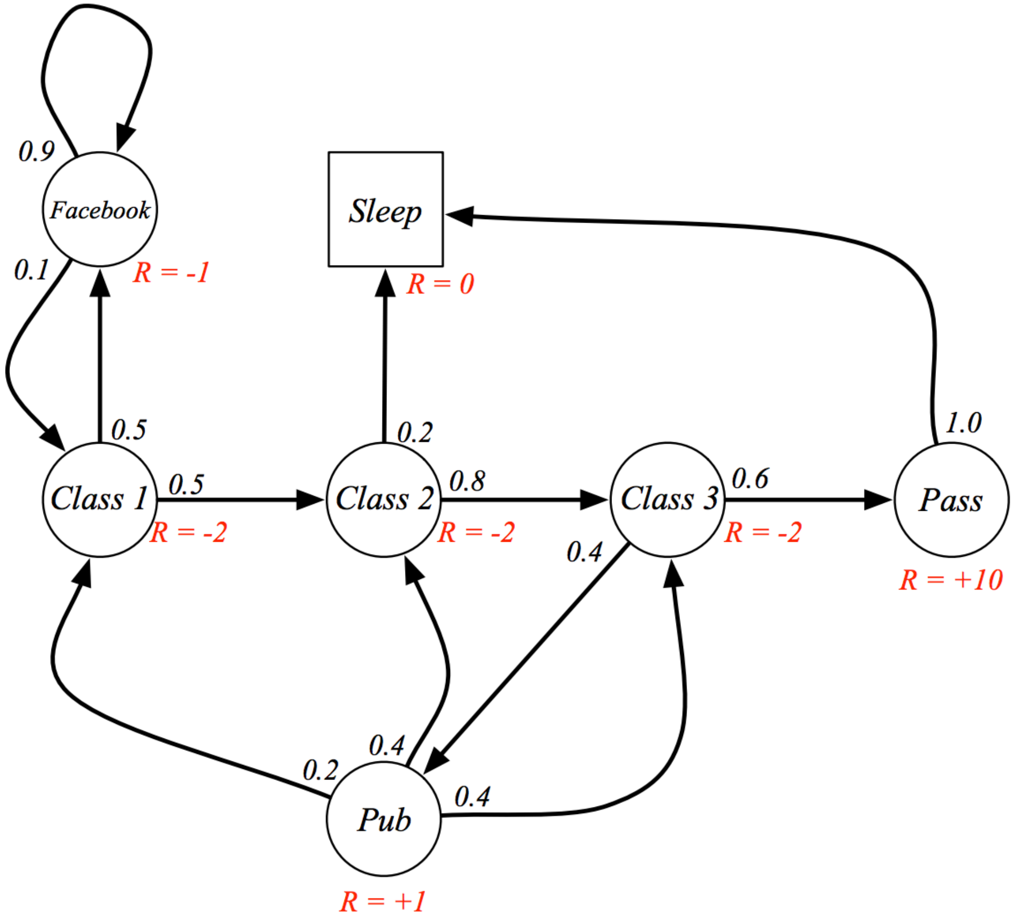
- R is a reward function, $R = \mathbb{E}\left[R_{t+1} \mid S_t = s\right]$
- γ is a discount factor, $\gamma \in [0, 1]$

Example: Mars Rover MRP

- Reward: +1 in S_1, +10 in S_5, 0 in the other states
- Discount factor $\gamma = 0.5$



Example: Student MRP



Reward over Multiple Transitions

- Return
  - Total discounted sum of rewards from time step *t*

Definition: The return G_t is the total discounted reward from time-step t

$$G_t = \underbrace{R_{t+1}}_{\text{Immediate reward}} + \underbrace{\gamma R_{t+2} + \cdots}_{\text{Discount sum of future reward}} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Discount factor γ is used
  - the present value of future rewards

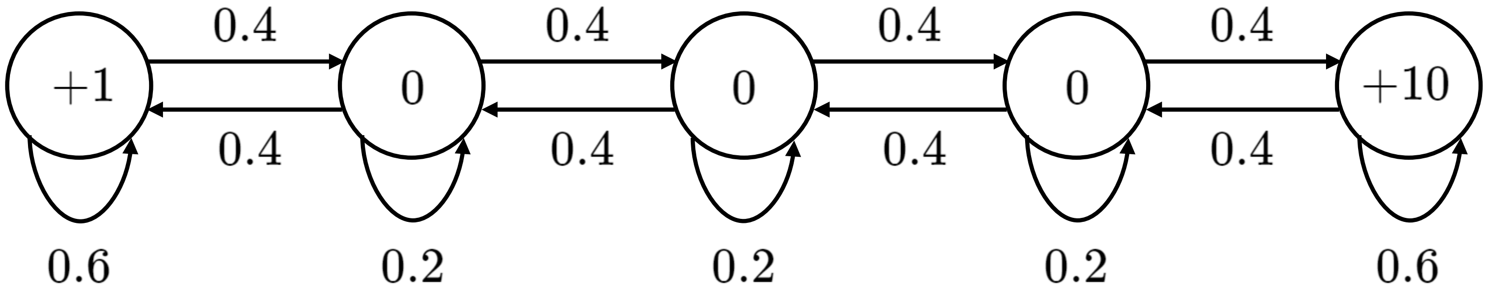
Discount factor γ

- It is reasonable to maximize the sum of rewards.
- It is also reasonable to prefer rewards now to rewards later.
- One solution: values of rewards decay exponentially
- Mathematically convenient (avoid infinite returns and values)

- Humans often act as if there's a discount factor $\gamma < 1$
  - $\gamma = 0$: Only care about immediate reward
  - $\gamma = 1$: Future reward is as beneficial as immediate reward

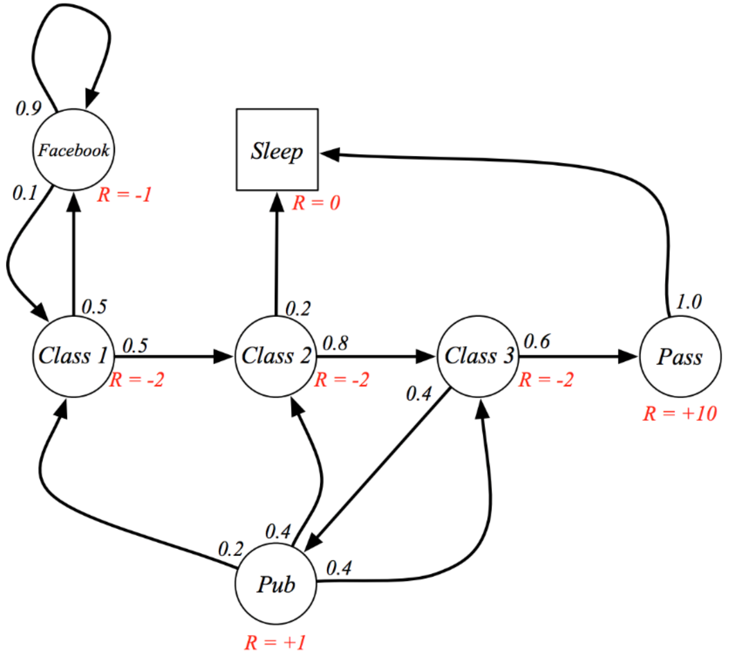
1	$\longrightarrow$	γ	$\longrightarrow$	γ^2
Worth Now		Worth Next Step		Worth In Two Steps

Example: Mars Rover MRP



- Reward: +1 in S_1, +10 in S_5, 0 in all other states
- Sample returns from sample episodes, $\gamma = 0.5$
  - $S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5 : 0 + (0.5 \times 0) + \left(0.5^2 \times 0\right) + \left(0.5^3 \times 10\right) = 1.25$
  - $S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_3 : 0 + (0.5 \times 0) + \left(0.5^2 \times 0\right) + \left(0.5^3 \times 0\right) = 0.0$
  - $S_2 \rightarrow S_3 \rightarrow S_2 \rightarrow S_2 : 0 + (0.5 \times 0) + \left(0.5^2 \times 0\right) + \left(0.5^3 \times 1\right) = 0.125$

Example: Student MRP Returns



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

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

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

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

- Expected return from starting from state s

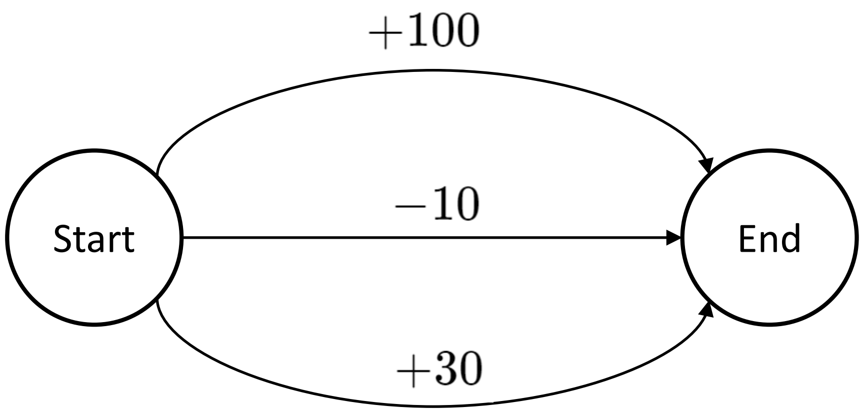
$$
\begin{aligned}
v(s) &= \mathbb{E}[G_t \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s]
\end{aligned}
$$

Computing Value Function of MRP (Naïve)

- Generate a large number of episodes and compute the average return

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

- Example



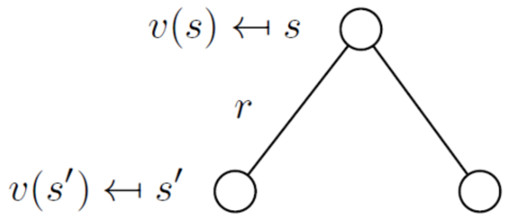
$$v(\text{Start}) = \frac{100 - 10 + 30}{3} = +40$$

Computing Value Function of MRP (Smart and Efficient)

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

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

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

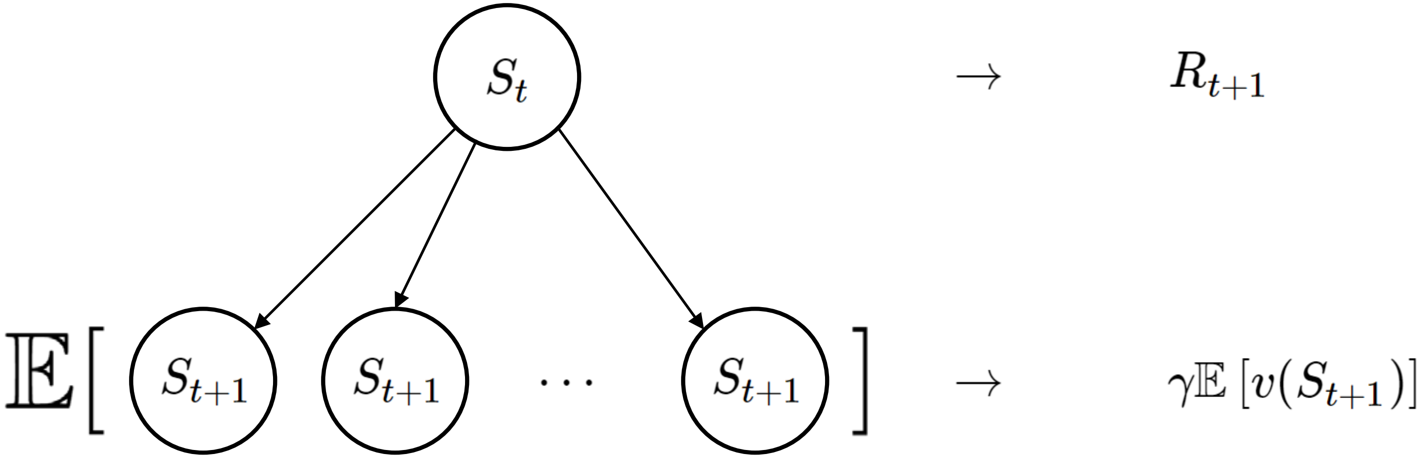


$$\boxed{v(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} v(s')}$$

Computing Value Function of MRP (Smart and Efficient)

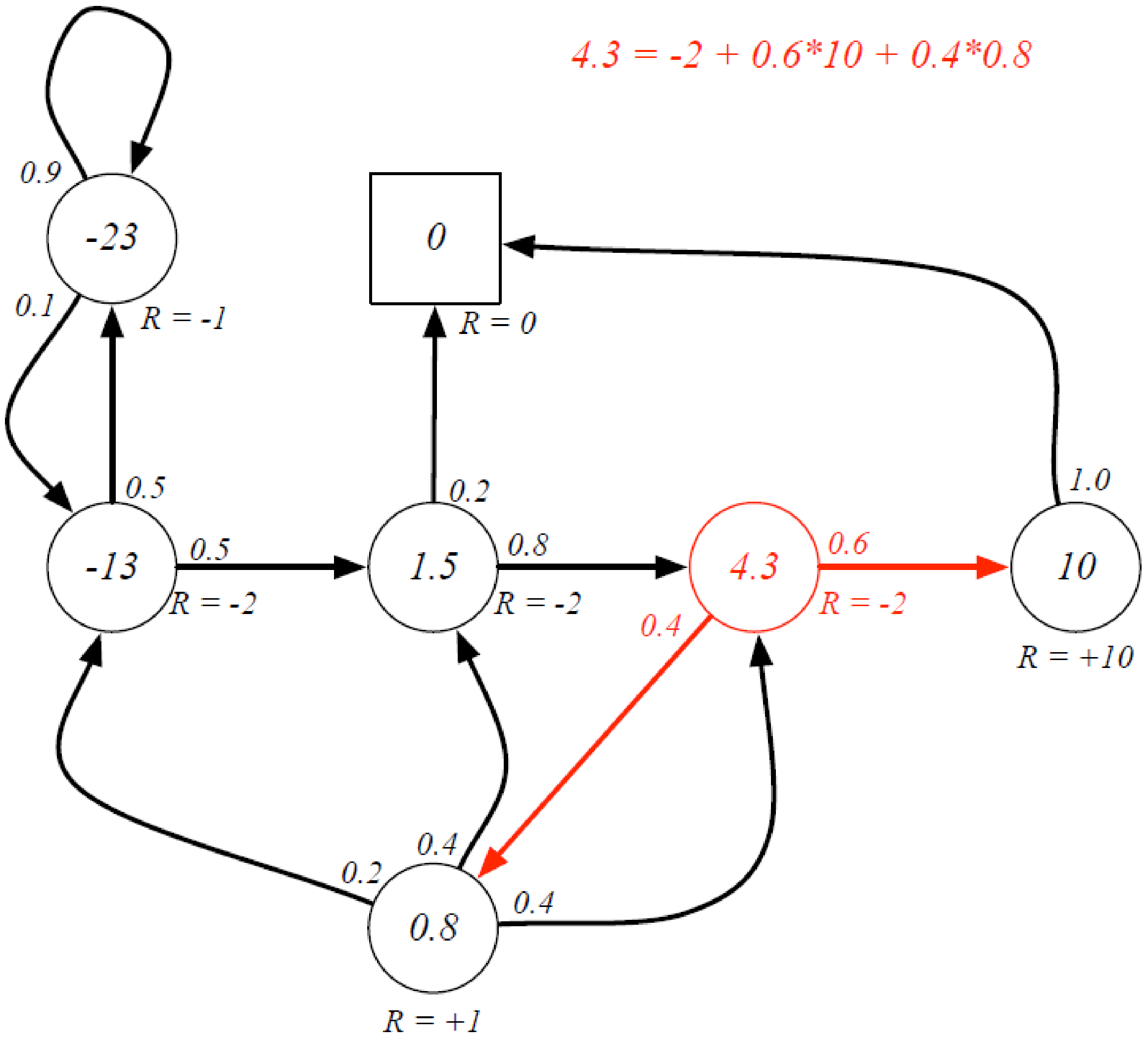
- Bellman Equations for MRP

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



$$v(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} v(s')$$

Example: Bellman Equation for Student MRP



Bellman Equation in Matrix Form

$$v(s) = R + \gamma \sum_{s' \in S} P_{ss'} v(s') \qquad \forall s$$

- The Bellman equation can be expressed concisely using matrices,

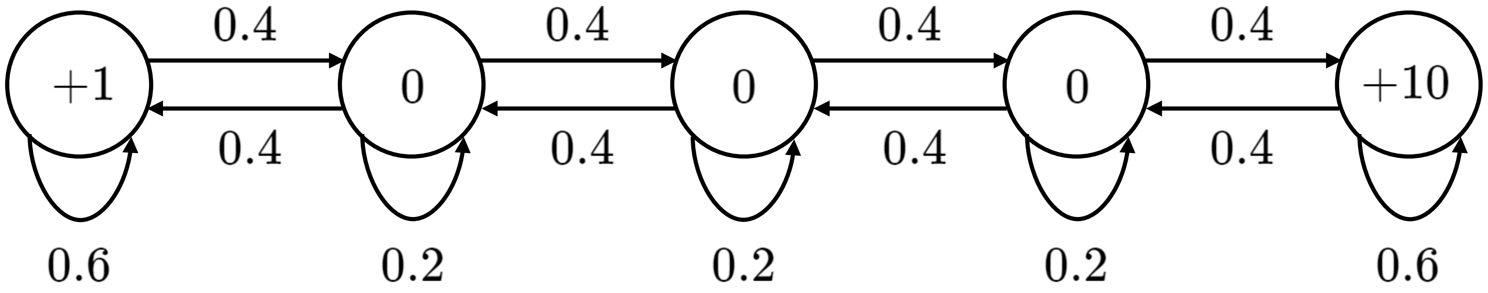
$$v = R + \gamma P v$$

- v is a column vector with one entry per state

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

Example: Mars Rover MRP

- Reward: +1 in S_1, +10 in S_5, 0 in the other states
- Discount factor $\gamma = 0.5$



$$v = R + \gamma P v$$

$$\begin{bmatrix} v(1) \\ v(2) \\ v(3) \\ v(4) \\ v(5) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} + 0.5 \begin{bmatrix} 0.6 & 0.4 & 0.0 & 0.0 & 0.0 \\ 0.4 & 0.2 & 0.4 & 0.0 & 0.0 \\ 0.0 & 0.4 & 0.2 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.4 & 0.2 & 0.4 \\ 0.0 & 0.0 & 0.0 & 0.4 & 0.6 \end{bmatrix} \begin{bmatrix} v(1) \\ v(2) \\ v(3) \\ v(4) \\ v(5) \end{bmatrix}$$

Solving the Bellman Equation

- Analytic solution for value function
- The Bellman equation is a linear equation
- It can be solved directly:

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

- Direct solution only possible for small MRP
- Computational complexity is $O(n^3)$ for n states

Iterative Algorithm for Value Function

- There are many *iterative* methods for large MRP
  - Dynamic programming
  - Monte-Carlo simulation
  - Temporal-difference learning

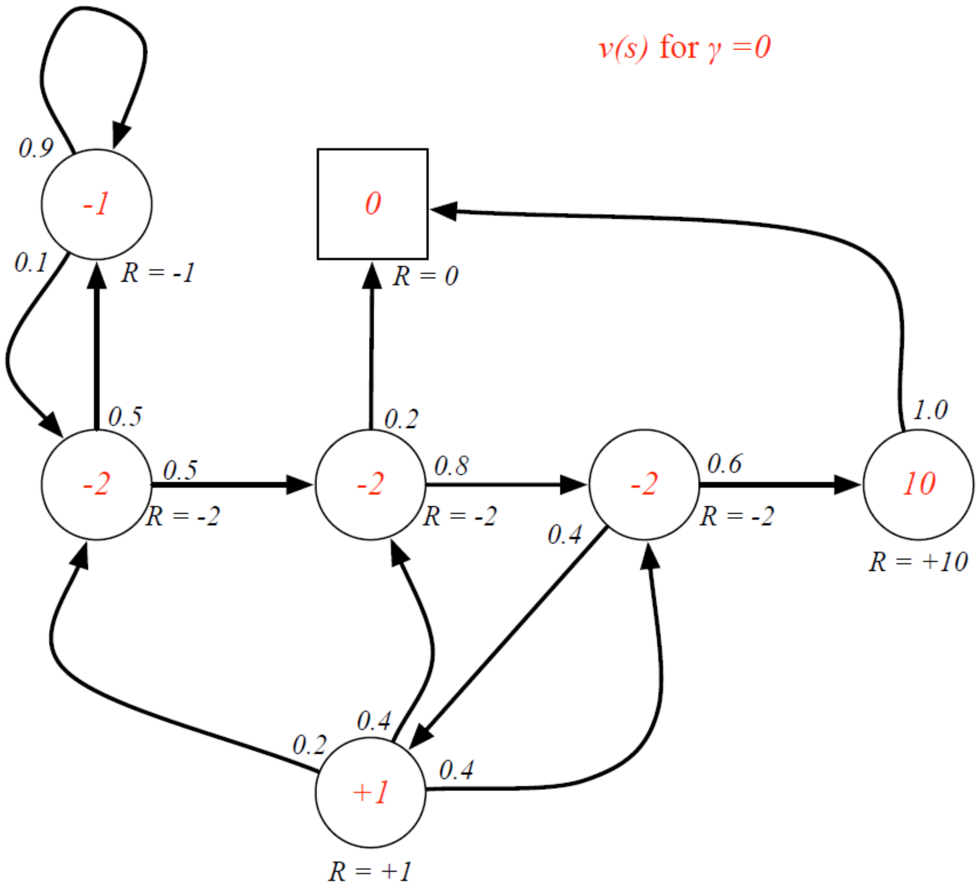
- Iterative algorithm for value function (Value Iteration)

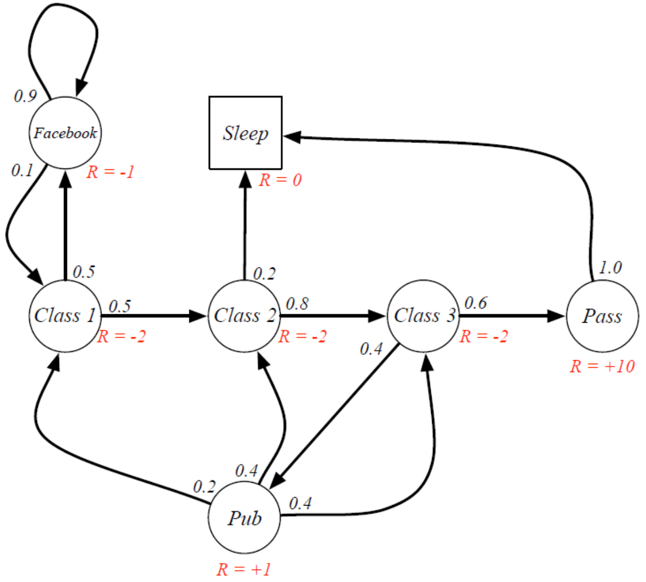
  - Initialize $V_1(s)$ for all s
  - For $k = 1$ until convergence
    - For all s in S

$$v_{k+1}(s) \longleftarrow R(s) + \gamma \sum_{s' \in S} p\left(s' \mid s\right) v_k\left(s'\right)$$

- Computational complexity: $O\left(n^2\right)$ for each k

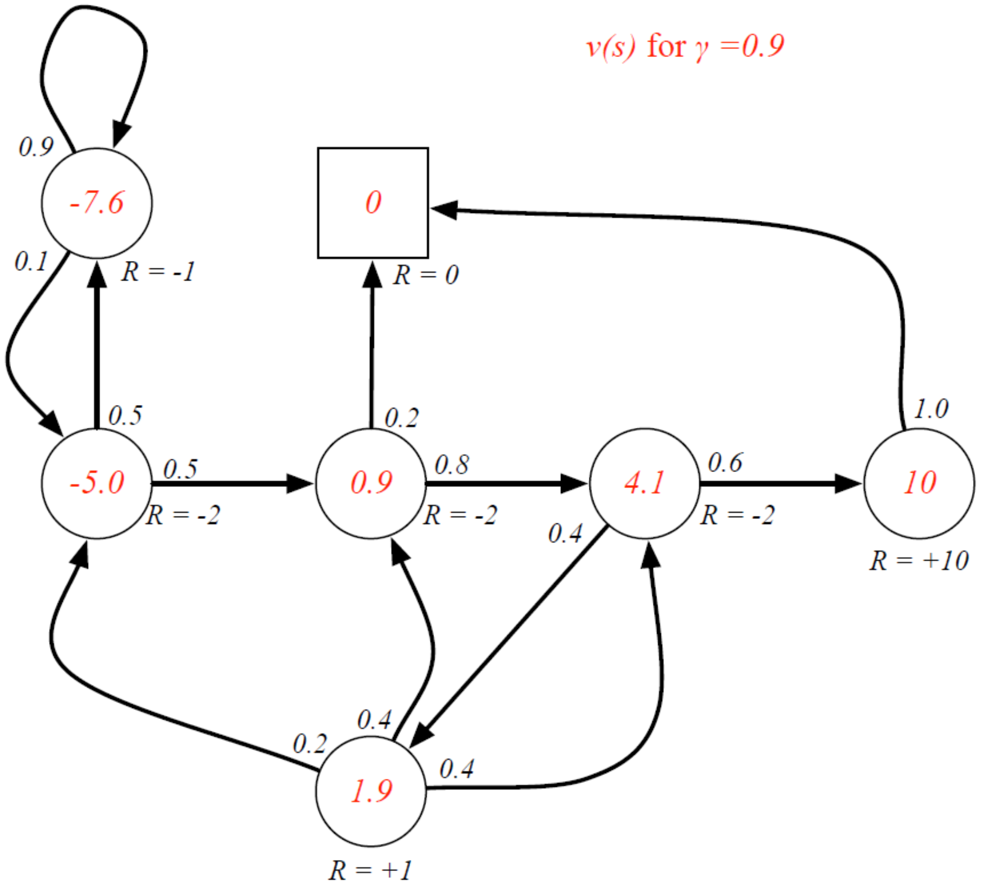
Value Function for Student MRP (1/3)

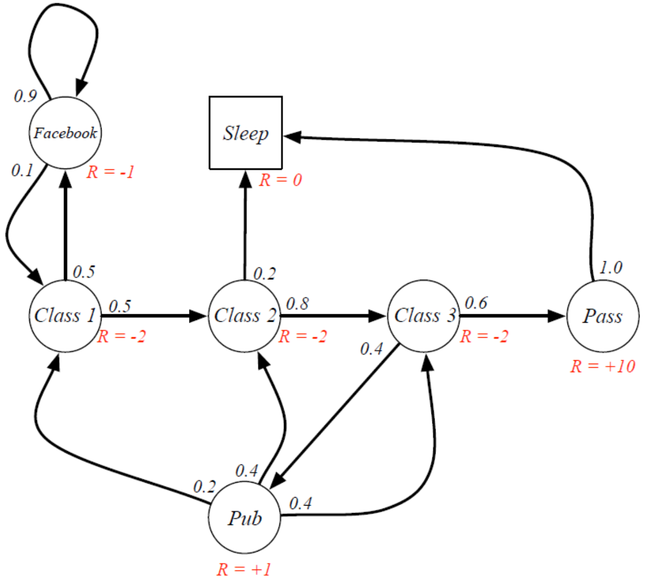




Value Function for Student MRP (2/3)

$v(s)$ for $\gamma = 0.9$





Value Function for Student MRP (3/3)