Марковский процесс принятия решений

А.Г. Трофимов
к.т.н., доцент, НИЯУ МИФИ

lab@neuroinfo.ru
http://datalearning.ru

Курс “Статистическая обработка временных рядов”

Сентябрь 2018

Markov Models

Markov model of a system assumes its Markov property, i.e. that its future states depend only on the current state, not on the events that occurred before it

Markov chain is the simplest Markov model when the state of a system is modelled as a random variable that changes through time (discrete or continuous)

	System state is fully observable	**System state is partially observable**
System is autonomous	Markov chain	Hidden Markov model
System is controlled	Markov decision process	Partially observable Markov decision process

Markov Reward Process

Markov Reward Process (MRP) is a Markov chain (discrete-time or continuous-time) with an associated reward to each state

Let $X = \{X_n, n = 0, 1, ...\}$ is a Markov chain with state space $S = \{s_1, ..., s_k\}$ and $R(s_1), ..., R(s_k)$ are rewards of states $s_1, ..., s_k$

The MRPs were developed in Ronald A. Howard's book "Dynamic Programming and Markov Processes" (1960)

Notes on rewards:

- The reward is a function of state $R : S \to \mathbb{R}$
- The rewards $R(s_1), ..., R(s_k)$ can be determined values or random variables
- The rewards represent "goodness" of states
- $R(s_i)$ is the immediate reward, it characterizes the reward that will be given immediately as soon as the chain stay on state s_i, $i = 1, ..., k$

Random Reward Signal

Each sample path $x_0, ..., x_n$ of the Markov chain $X = \{X_n, n = 0, 1, ...\}$ gives associated sample of rewards $r_1, ..., r_n$ that can be viewed as a sample path of a random reward signal $\{R_n, n = 1, 2, ...\}$

If rewards $R(s_1), ..., R(s_k)$ are random, then each r_n is an observation of random variable $R(x_n)$. If rewards $R(s_1), ..., R(s_k)$ are deterministic, then each reward $r_n = R(x_n)$

The cumulative (total) reward up to step n:

$$C_n = \sum_{i=1}^{n} R_i$$

The total reward $c_n = \sum_{i=1}^{n} r_i$ represents the "goodness" of the sample path $x_0, ..., x_n$

Reward Function of MRP

Definition

The reward function $\rho(s) \in \mathbb{R}$ of state $s \in S$ of an MRP X is the expected reward received upon leaving state s:

$$\rho(s) = \mathrm{M}[R_{n+1} \mid X_n = s]$$

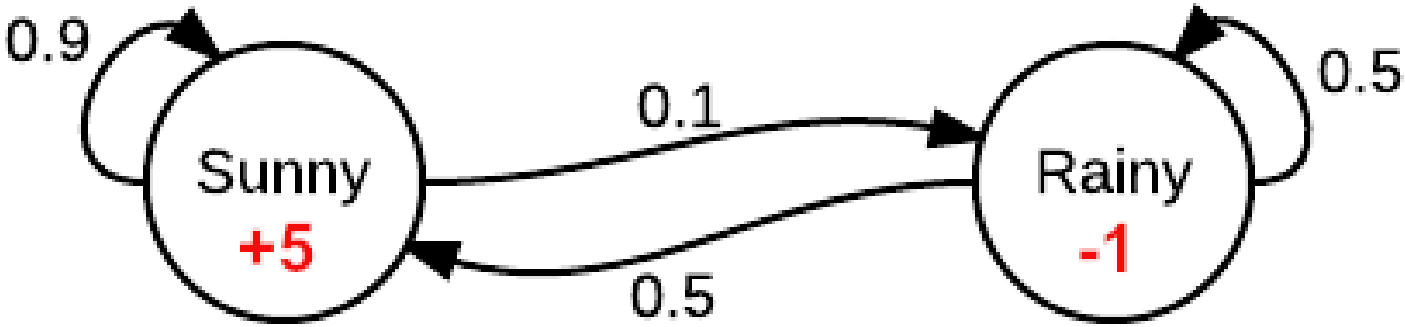
Reward $R(s_i)$ of each state s_i, $i = 1, ..., k$, can be random, but the value of reward function $\rho(s_i)$ is deterministic, it is expectation of all rewards that can be received upon leaving state s_i:

$$\rho(s_i) = \mathrm{M}[R_{n+1} \mid X_n = s_i] = \sum_{j=1}^{k} \mathrm{M}[R(s_j)]p_{ij}$$

where p_{ij} is a transition probability from s_i to s_j

The reward function ρ is the mapping from state space S to the expected immediate rewards, $\rho : S \to \mathbb{R}$

Markov Reward Process. Example



States: $S = \{s_1 : \text{Sunny}, s_2 : \text{Rainy}\}$
Rewards of states: $R(s_1) = 5$, $R(s_2) = -1$
Expected immediate reward (reward function) of states:
$\rho(s_1) = 0.9 \cdot 5 + 0.1 \cdot (-1) = 4.4$
$\rho(s_2) = 0.5 \cdot 5 + 0.5 \cdot (-1) = 2.0$
A sample path: $s_2, s_1, s_1, s_2, s_1, s_2, ...$
Sample of rewards: $5, 5, -1, 5, -1, ...$
Sample of total rewards: $5, 10, 9, 14, 13, ...$

n-Step Expected Reward

All rewards $R_1, ..., R_n$ are random variables, they depend on the random trajectory $X_0, ..., X_n$ of the Markov chain up to time step n

Assume at time step n the Markov chain X has state probability distribution $\pi^{(n)}$:

$$\pi^{(n)} = \left(\pi_1^{(n)}, ..., \pi_k^{(n)}\right) = (P(X_n = s_1), ..., P(X_n = s_k))$$

The expected reward at time step n:

$$\mathrm{M}[R_n] = \sum_{i=1}^{k} \mathrm{M}[R(s_i)] P(X_n = s_i) = \pi^{(n)} r = \pi^{(0)} P^n r$$

where $r = (\mathrm{M}[R(s_1)], ..., \mathrm{M}[R(s_k)])^T$ is the vector or expected state rewards, P is the transition probability matrix and $\pi^{(0)}$ is the initial state distribution vector of the Markov chain X

Return of MRP

Definition

The return G_n of an MRP at time step n is a specific function of future rewards:

$$G_n = G(R_{n+1}, ..., R_T)$$

where T is a final time step

In the simplest case, the return at time step n is a sum of future rewards, i.e. future cumulative reward:

$$G_n = R_{n+1} + R_{n+2} + ... + R_T$$

The return G_n depends on the current state X_n and represents the future cumulative reward that can be received from the state X_n

The final time step T is called time horizon

Time Horizon

The time horizon can be determined or random, finite or infinite

If the final time step T is finite then the sample paths of Markov chain up to step T are called episodes (or trials)

Types of time horizon:

- Finite
  The process terminates after a particular fixed number of time steps T
- Indefinite
  The process can last arbitrarily long but must eventually terminate in an absorbing state of Markov chain with zero reward
- Infinite
  The process does not terminate (e.g. if the Markov chain is periodic)

Discounted Return

For infinite time horizon the future cumulative reward is infinite, there is no sense to optimize it

The discounted return:

$$G_n = R_{n+1} + \gamma R_{n+2} + \gamma^2 R_{n+3} + ...$$

where γ is a parameter, $0 \leq \gamma \leq 1$, called the discount rate

The discount rate γ determines the present value of future rewards: a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately

The discounting avoids infinite returns for infinite-horizon MRP

If the rewards of all states are constant R_0, then the discounted return is finite:

$$G_n = R_0(1 + \gamma + \gamma^2 + ...) = \frac{R_0}{1 - \gamma}$$

Discount Rates of MRP

Possible values of the discount rate:

- $\gamma = 0$ ("myopic" return)

$$G_n = R_{n+1}$$

The return G_n at time step n depends only on the next successor reward

- $\gamma = 1$ ("far-sighted" return)

$$G_n = R_{n+1} + R_{n+2} + ...$$

All future rewards are worth equally, the value of reward doesn't depend on the time when it will be received

- $0 < \gamma < 1$

$$G_n = R_{n+1} + \gamma R_{n+2} + \gamma^2 R_{n+3} + ...$$

The further the reward is received, the less valuable it is

State-Value Function of MRP

Definition

The value $v(s) \in \mathbb{R}$ of state $s \in S$ of an MRP X is the expected return starting from state s:

$$v(s) = \mathrm{M}[G_n \mid X_n = s]$$

The return G_n is a random variable, it depends on the random future trajectory $X_{n+1}, X_{n+2}, \ldots$ of the Markov chain

The value $v(s)$ is not random, it is expectation of all possible returns that can be received when starting from state s

The value $v(s)$ characterizes state s and doesn't depend on the current time step n

The state-value function v is the mapping from state space S to the expected returns (state values), $v : S \to \mathbb{R}$

Bellman Equation for MRP

The state-value function of MRP with discounted return:

$$
\begin{aligned}
v(s_i) &= \mathrm{M}[G_n \mid X_n = s_i] \\
&= \mathrm{M}[R_{n+1} + \gamma R_{n+2} + \gamma^2 R_{n+3} + \ldots \mid X_n = s_i] \\
&= \mathrm{M}[R_{n+1} + \gamma(R_{n+2} + \gamma R_{n+3} + \ldots) \mid X_n = s_i] \\
&= \mathrm{M}[R_{n+1} + \gamma G_{n+1} \mid X_n = s_i] \\
&= \mathrm{M}[R_{n+1} \mid X_n = s_i] + \gamma \mathrm{M}[G_{n+1} \mid X_n = s_i] \\
&= \rho(s_i) + \gamma \mathrm{M}[G_{n+1} \mid X_{n+1} = S_j, X_n = s_i] \\
&= \rho(s_i) + \gamma \mathrm{M}[v(S_j) \mid X_n = s_i] \\
&= \rho(s_i) + \gamma \sum_{j=1}^{k} p_{ij} v(s_j), \quad i = 1, \ldots, k
\end{aligned}
$$

This equation is called the Bellman equation for MRP

Bellman Equation in Matrix Form

The Bellman equation:

$$v(s_i) = \rho(s_i) + \gamma \sum_{j=1}^{k} p_{ij} v(s_j), \quad i = 1, ..., k$$

The value of state s_i is the immediate expected reward we get upon leaving that state, plus a discounted weighted average value of next possible states, where the weights are the transition probabilities to that states

In matrix form:

$$v = \rho + \gamma P v$$

$$\begin{pmatrix} v(s_1) \\ ... \\ v(s_k) \end{pmatrix} = \begin{pmatrix} \rho(s_1) \\ ... \\ \rho(s_k) \end{pmatrix} + \gamma \begin{pmatrix} p_{11} & ... & p_{1k} \\ ... & ... & ... \\ p_{k1} & ... & p_{kk} \end{pmatrix} \begin{pmatrix} v(s_1) \\ ... \\ v(s_k) \end{pmatrix}$$

Solution of Bellman Equation

The Bellman equation is linear w.r.t. v and can be solved directly:

$$v = \rho + \gamma P v$$

$$(I - \gamma P)v = \rho$$

$$v = (I - \gamma P)^{-1}\rho$$

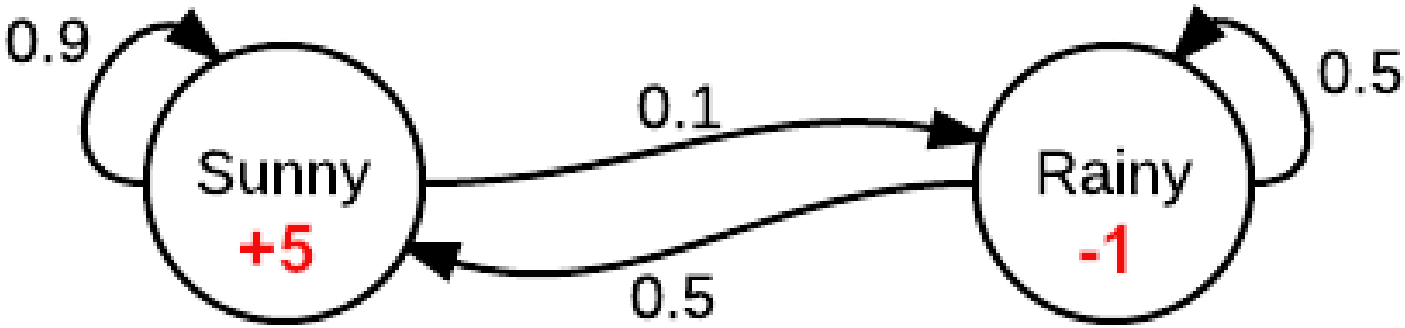
In practice, it can be solved only for small k, because of the inverse operation (complexity $O(k^3)$)

For discount rate $\gamma = 0$:

$$v = I^{-1}\rho = \rho$$

If the immediate reward only matters, than the state-value function is equal to reward function

Example. State-Value Function



States: $S = \{s_1 : \text{Sunny}, s_2 : \text{Rainy}\}$

Transition probability matrix: $P = \begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix}$

Rewards of states: $r = (5, -1)^T$

Discount rate: $\gamma = 0.9$

Reward function: $\rho = Pr = (4.4, 2.0)^T$

State-value function:

$$v = (I - \gamma P)^{-1}\rho = \begin{pmatrix} 0.19 & -0.09 \\ -0.45 & 0.55 \end{pmatrix}^{-1} \begin{pmatrix} 4.4 \\ 2.0 \end{pmatrix} \approx \begin{pmatrix} 40.6 \\ 36.9 \end{pmatrix}$$

Reward Signal vs Value Function

Reward signal $\{R_n, n = 1, 2, ...\}$ and state-value function $v(s)$ are the central elements of Markov reward processes

- The reward signal defines a "goodness" of MRP's sample path
- Whereas the reward signal indicates what is good in an immediate sense, a value function specifies what is good in the long run
- The value of a state is the total amount of reward we can expect to accumulate over the future, starting from that state while rewards determine the immediate, intrinsic desirability of MRP's states
- States can have low rewards but high values and vice-versa
- Rewards are given directly by the state of the chain, but values must be computed as a solution of MRP's Bellman equation (solution of the MRP)

Markov Decision Process

Markov Decision Process (MDP) is a Markov reward process with actions

MDP is a control process that means that state transitions are supervised by a decision maker

At each time step, the process is in some state s, and the decision maker choose an action a that is available in state s. The process responds at the next time step by randomly (w.r.t. transition probabilities) moving into a new state s', and giving a corresponding reward r

If only one action is available for each state, a Markov decision process reduces to Markov reward process

The Markov property of MDP: given current state s and action a, the future states of MDP are independent of all previous states and actions taken

Definition of MDP

MDP is characterized by:

- Finite set of states S
- Finite set of actions A

  Actions can be associated to each state separately, in this case the MDP is characterized by sets $A(s)$, $s \in S$
- State transition probabilities P

  The state transition probabilities in MDP are conditioned on a chosen action, i.e. $P : S \times A \times S \to [0, 1]$

  Transition probabilities matrices can be associated to each action separately: P_a, $a \in A$
- Rewards $R : S \times A \times S \to \mathbb{R}$

  The reward is associated to transition from state $s \in S$ to state $s' \in S$ due to action $a \in A$

  The reward also can be received due to action $a \in A$ taken in state $s \in S$, in this case the reward is $R : S \times A \to \mathbb{R}$

MDP vs MRP. State Transition Probabilities

Let $X = \{X_n, n = 0, 1, ...\}$ be a Markov chain with state space $S = \{s_1, ..., s_k\}$

For MRP:

The state transition probabilities are associated to the pair of states:

$$p_{ij} = P(X_{n+1} = s_j \mid X_n = s_i), \quad i, j = 1, ..., k$$

For MDP:

The state transition probabilities are associated to the triples (state, action, state):

$$p_{a,ij} = P(X_{n+1} = s_j \mid X_n = s_i, A_n = a), \quad i, j = 1, ..., k$$

$p_{a,ij}$ is a transition probability from s_i to s_j due to action $a \in A(s_i)$

MDP vs MRP. Rewards

For MRP:

- The rewards are associated to states
  $R(s_i)$ characterizes the reward that will be given immediately as soon as the chain stay on state s_i, $i = 1, ..., k$

For MDP:

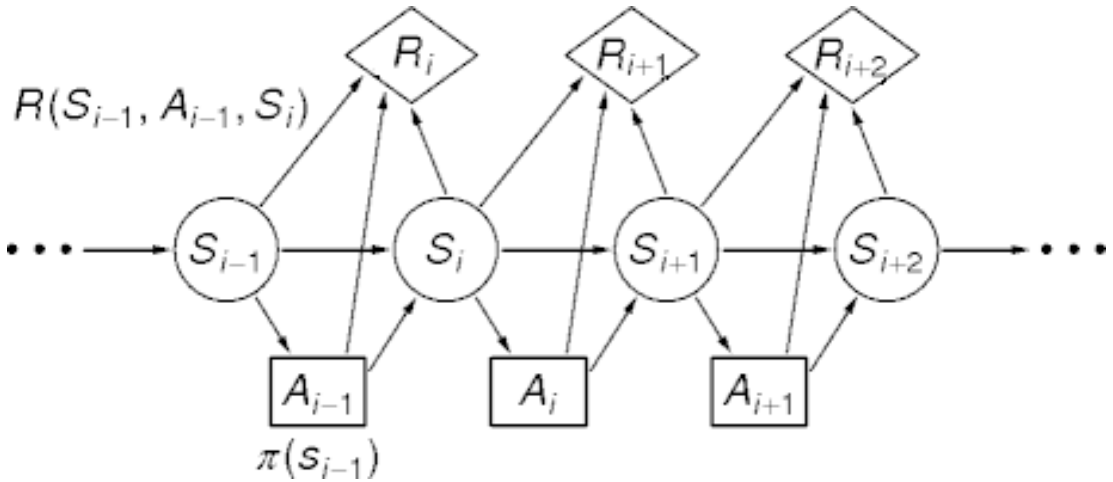
- The rewards are associated to triples (state, action, state)
  $R(s_i, a, s_j)$ characterizes the reward that will be given immediately as soon as decision maker takes action $a \in A$ in state s_i and the chain jumps to state s_j, $i, j = 1, ..., k$
- The rewards are associated to pairs (state, action)
  $R(s_i, a)$ characterizes the reward that will be given immediately as soon as decision maker takes action $a \in A$ in state s_i, $i = 1, ..., k$

MDP Transition Diagram

There are two kinds of nodes: state nodes and action nodes in MDP transition diagram

There is a state node for each state $s \in S$ and an action node for each state-action pair $(s, a) \in A(s)$

Starting in state s and taking action a the process moves along the line from state node s to action node (s, a). Then the process responds with a transition to the next state's node s' via one of the arrows leaving action node (s, a) and receives a reward $R(s, a, s')$

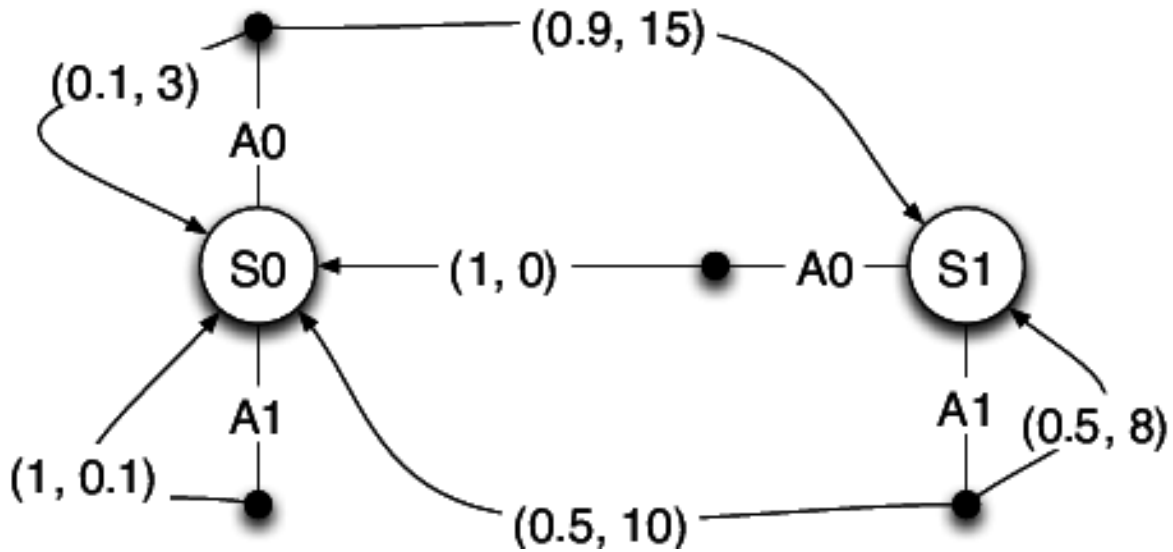


MDP Transition Diagram. Example 1

The connection from state node s_i to action node a is marked by action a. The connection from action node a to the next state s_j is marked by a pair $(p_{a,ij}, R(s_i, a, s_j))$, $i, j = 1, ..., k$

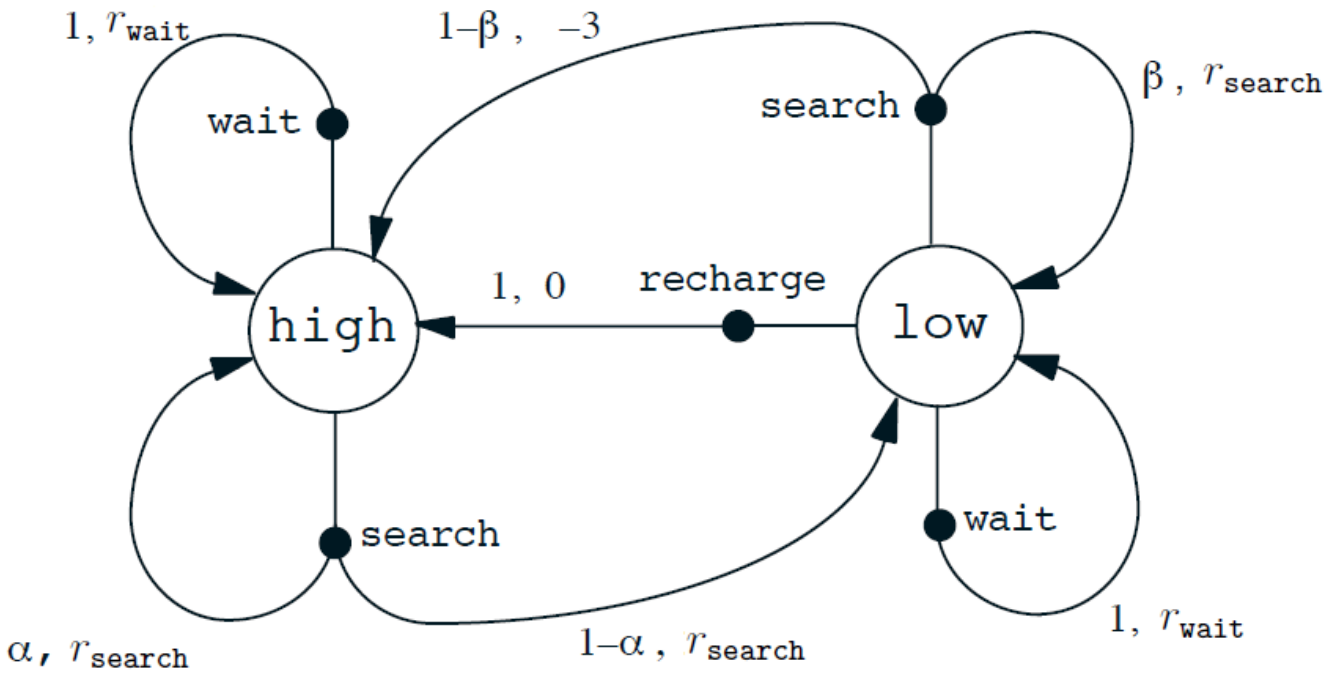
The sum of elements in each row of matrix P_a is equal to 1:

$$\sum_{j=1}^{k} p_{a,ij} = \sum_{j=1}^{k} P(s_j \mid s_i, a) = 1, \quad i = 1, ..., k$$



MDP Transition Diagram. Example 2

Transition diagram for the recycling robot example*



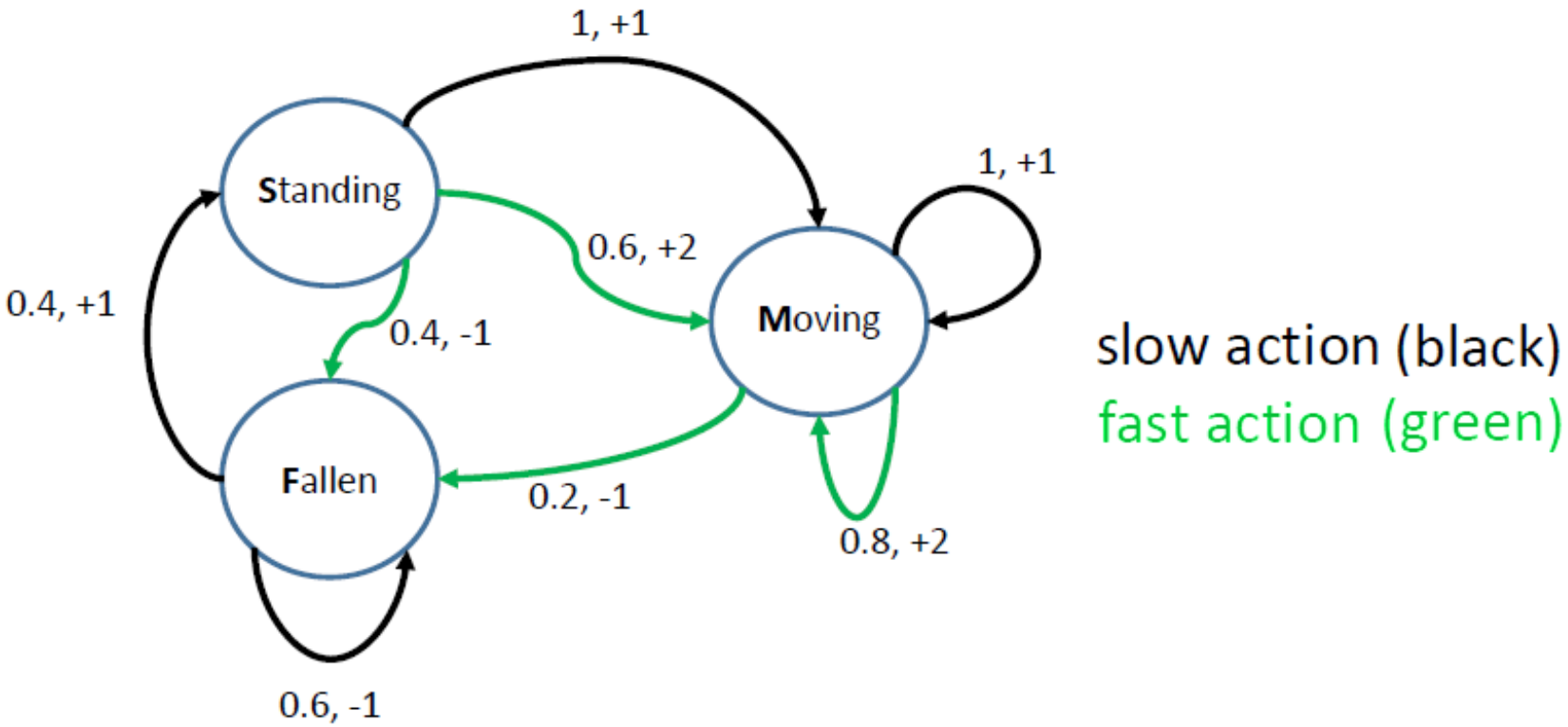
$S = \{\text{high battery}, \text{low battery}\}$

$A(\text{high}) = \{\text{search}, \text{wait}\},\ A(\text{low}) = \{\text{search}, \text{wait}, \text{recharge}\}$

*R.S.Sutton, A.G.Barto (1998). Reinforcement learning: An introduction. MIT press.

MDP Transition Diagram. Example 3

Transition diagram for a robot trying to walk



$S = \{\text{Standing}, \text{Moving}, \text{Fallen}\}$, $A = \{\text{slow action}, \text{fast action}\}$

Action nodes are substituted by colors to simplify the diagram

Random Reward Signal

Each sample path $x_0, ..., x_n$ of the Markov chain $X = \{X_n, n = 0, 1, ...\}$ driven by actions $a_0, ..., a_{n-1}$ gives associated sample of rewards $r_1, ..., r_n$ that can be viewed as a sample path of a random reward signal $\{R_n, n = 1, 2, ...\}$

The cumulative (total) reward up to step n:

$$C_n = \sum_{i=1}^{n} R_i$$

The total reward $c_n = \sum_{i=1}^{n} r_i$ represents the "goodness" of the sample path $x_0, ..., x_n$ and actions $a_0, ..., a_{n-1}$ taken by decision maker

Example:
The sample path
[Standing]–fast–[Moving]–fast–[Moving]–fast–[Fallen]
gives cumulative reward $c = 2 + 2 - 1 = 3$

Reward Function of MDP

Definition

The reward function $\rho(s, a) \in \mathbb{R}$ of state $s \in S$ and action $a \in A(s)$ is the expected reward received upon leaving state s due to action a:

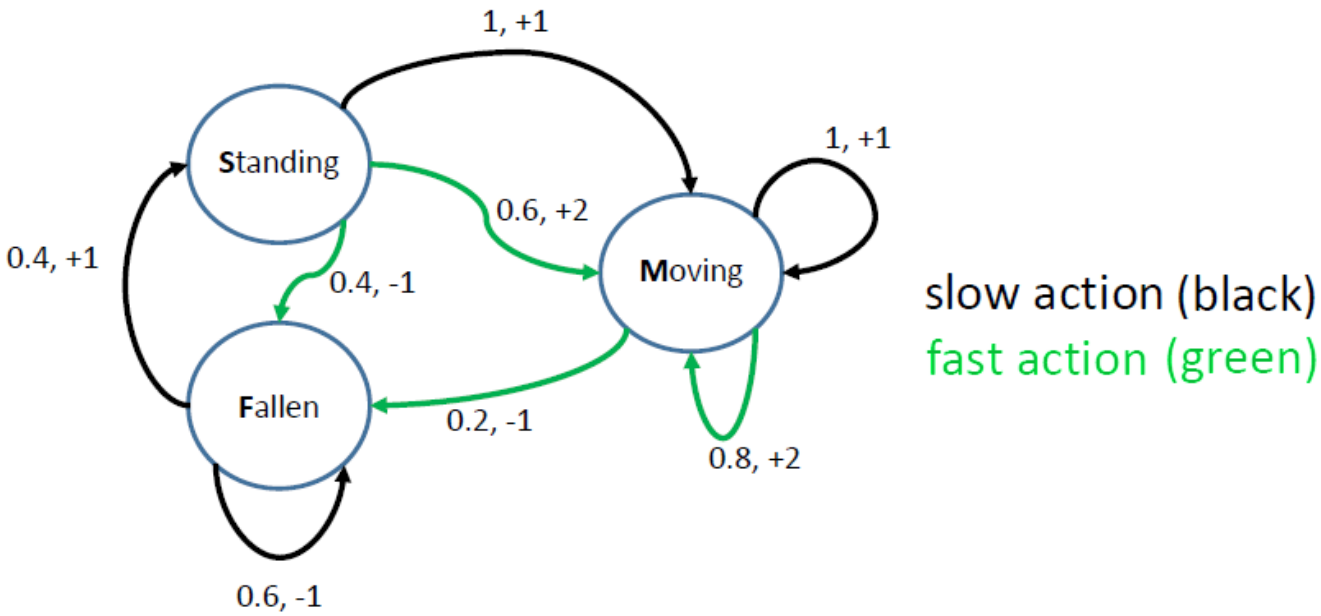
$$\rho(s, a) = \mathrm{M}[R_{n+1} \mid X_n = s, A_n = a]$$

Rewards $R(s_i, a, s_j)$, $a \in A(s_i)$, can be random, but the value of reward function $\rho(s_i, a)$ is deterministic, it is expectation of all rewards that can be received upon leaving state s_i due to action a:

$$\rho(s_i, a) = \mathrm{M}[R_{n+1} \mid X_n = s_i, A_n = a] = \sum_{j=1}^{k} \mathrm{M}[R(s_i, a, s_j)] p_{a,ij}$$

The reward function ρ is the mapping from state-action space to the expected immediate rewards, $\rho : S \times A \to \mathbb{R}$

Reward Function. Example



Expected immediate reward (reward function):

$\rho(\text{Standing, slow action}) = 1$

$\rho(\text{Standing, fast action}) = 0.6 \cdot 2 + 0.4 \cdot (-1) = 0.8$

$\rho(\text{Moving, slow action}) = 1$

$\rho(\text{Moving, fast action}) = 0.8 \cdot 2 + 0.2 \cdot (-1) = 1.4$

$\rho(\text{Fallen, slow action}) = 0.4 \cdot 1 + 0.6 \cdot (-1) = -0.2$

$\rho(\text{Fallen, fast action}) = undefined$

Return of MDP

The return of MDP represents the future cumulative reward that can be received from state X_n

For finite time horizon:

$$G_n = R_{n+1} + R_{n+2} + ... + R_T$$

For infinite time horizon:

$$G_n = R_{n+1} + \gamma R_{n+2} + \gamma^2 R_{n+3} + ...$$

where γ is discount rate, $0 \leq \gamma \leq 1$

The return G_n depends on the current state X_n and actions that will be taken in time moment n and thereafter

Policy

Definition

A policy π is a mapping from states to probabilities of selecting each possible action. If the decision maker is following policy π at time step $n \in \{0, 1, ...\}$, then

$$\pi(a|s) = P(A_n = a \mid X_n = s), \quad a \in A(s), \ s \in S$$

The policy π defines the a probability distribution over actions $a \in A(s)$ that the decision maker will choose for each state $s \in S$

If the policy

$$\pi(a|s) = \begin{cases} 1, & a = a^*(s), \\ 0, & \text{otherwise} \end{cases} \qquad \forall s \in S$$

then the decision maker takes deterministic decisions in each state, the policy π is called deterministic (otherwise stochastic)

Value of States

Definition

The value $v_\pi(s) \in \mathbb{R}$ of state $s \in S$ under policy π of an MDP X is the expected return starting in state s and following policy π thereafter:

$$v_\pi(s) = \mathrm{M}_\pi[G_n \mid X_n = s]$$

where G_n is the return of MDP, $\mathrm{M}_\pi[\cdot]$ denotes the expected value given that the decision maker follows policy π

The return G_n is a random variable, it depends on the random future trajectory of the Markov chain and therefore on the actions taken

The value $v_\pi(s)$ is not random, it is expectation of all possible returns that can be received when starting in state s and following policy π thereafter

Value of Actions

Definition

The value $q_\pi(s, a) \in \mathbb{R}$ of taking action $a \in A(s)$ in state $s \in S$ under policy π of an MDP X is the expected return starting in state s, taking action a, and thereafter following policy π:

$$q_\pi(s, a) = \mathrm{M}_\pi[G_n \mid X_n = s, A_n = a]$$

where G_n is the return of MDP, $\mathrm{M}_\pi[\cdot]$ denotes the expected value given that the decision maker follows policy π

The value $q_\pi(s, a)$ is not random, it's expectation of all possible returns that can be received when starting in state s, taking action a, and thereafter following the policy π

Values of states and values of actions in MDP can be defined only w.r.t. a policy π

State-Value and Action-Value Functions

The state value $v_\pi(s)$ characterizes state $s \in S$ w.r.t. a policy π and doesn't depend on the current time step n

The state-value function v_π for policy π is the mapping from state space S to the expected returns (state values):

$$v_\pi : S \to \mathbb{R}$$

The action value $q_\pi(s, a)$ characterizes action $a \in A(s)$ taken in state $s \in S$ w.r.t. a policy π and doesn't depend on the current time step n

The action-value function q_π for policy π is the mapping from space $S \times A$ to the expected returns (state values):

$$q_\pi : S \times A \to \mathbb{R}$$

Which is relation between state-value and action-value functions?

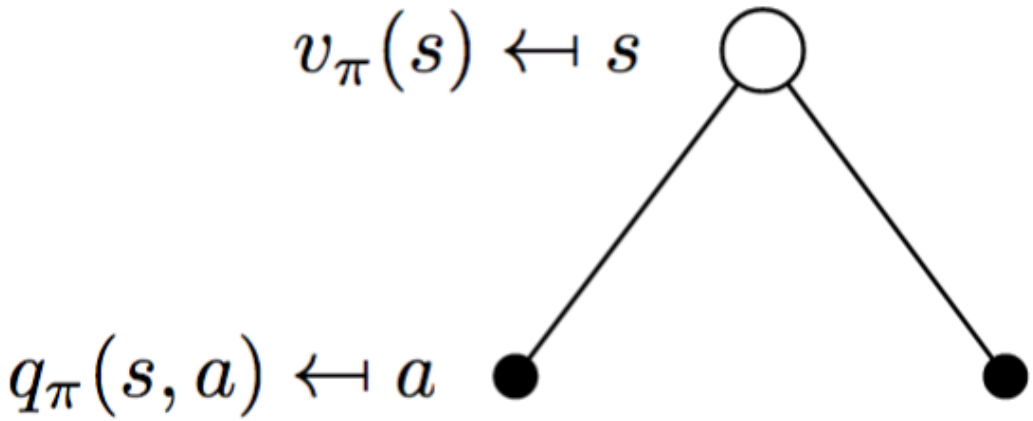
Look-Ahead Diagram for State-Value Function

State-value function:

$$v_\pi(s_i) = \mathrm{M}_\pi[G_n \mid X_n = s_i] = \sum_{a \in A(s_i)} \mathrm{M}_\pi[G_n \mid X_n = s_i, A_n = a]\pi(a|s_i)$$

$$= \sum_{a \in A(s_i)} q_\pi(s_i, a)\pi(a|s_i), \quad i = 1, ..., k$$

The value of state s_i w.r.t. policy π is the average of all available action-values weighted by the probabilities of actions under policy π



Expression for Action-Value Function

The action-value function of MRP with discounted return:

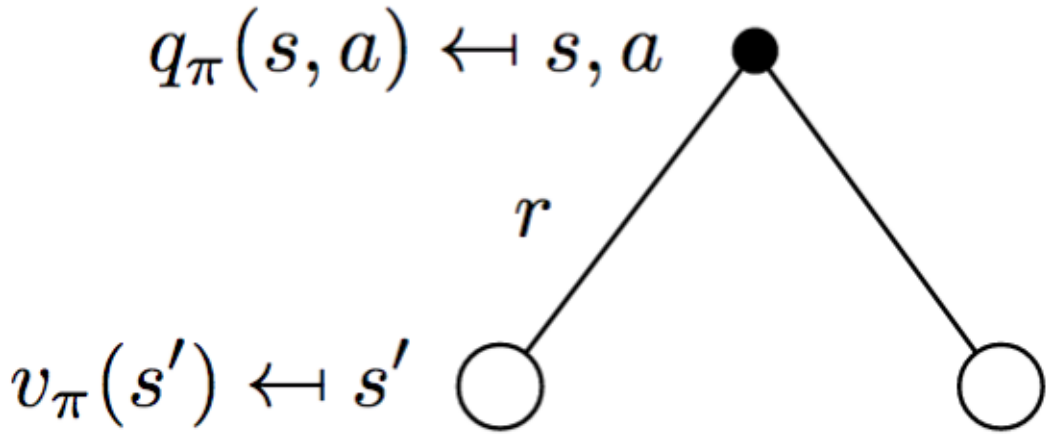
$$
\begin{aligned}
q_\pi(s_i, a) &= \mathrm{M}_\pi[G_n \mid X_n = s_i, A_n = a] \\
&= \mathrm{M}_\pi[R_{n+1} + \gamma R_{n+2} + \gamma^2 R_{n+3} + \ldots \mid X_n = s_i, A_n = a] \\
&= \mathrm{M}_\pi[R_{n+1} + \gamma(R_{n+2} + \gamma R_{n+3} + \ldots) \mid X_n = s_i, A_n = a] \\
&= \mathrm{M}_\pi[R_{n+1} + \gamma G_{n+1} \mid X_n = s_i, A_n = a] \\
&= \mathrm{M}[R_{n+1} \mid X_n = s_i, A_n = a] \\
&+ \gamma \mathrm{M}_\pi[G_{n+1} \mid X_n = s_i, A_n = a] \\
&= \rho(s_i, a) + \gamma \mathrm{M}_\pi[G_{n+1} \mid X_{n+1} = S_j, X_n = s_i, A_n = a] \\
&= \rho(s_i, a) + \gamma \mathrm{M}[v_\pi(S_j) \mid X_n = s_i, A_n = a] \\
&= \rho(s_i, a) + \gamma \sum_{j=1}^{k} p_{a,ij} v_\pi(s_j), \quad i = 1, \ldots, k
\end{aligned}
$$

Look-Ahead Diagram for Action-Value Function

Action-value function:

$$q_\pi(s_i, a) = \rho(s_i, a) + \gamma \sum_{j=1}^{k} p_{a,ij} v_\pi(s_j), \quad i = 1, ..., k$$

The value of action a in state s_i w.r.t. policy π is the immediate expected reward we get upon leaving that state, plus a discounted average value of next possible states w.r.t. policy π, weighted by the transition probabilities to that states due to action a

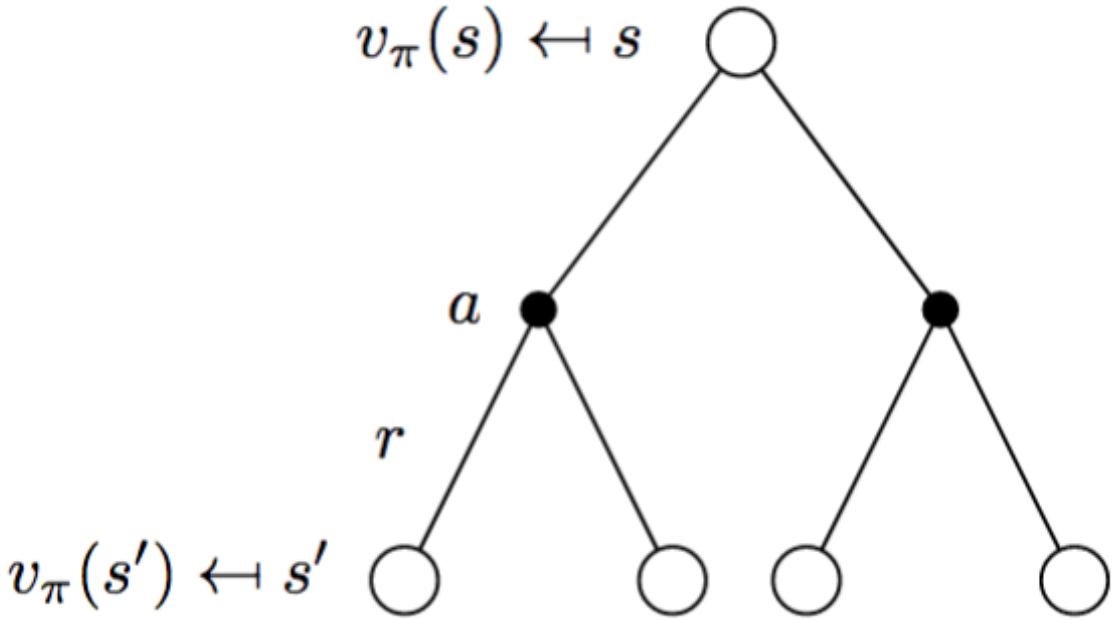


Bellman Equation for State-Value Function

Recurrent expression for state-value function:

$$v_\pi(s_i) = \sum_{a \in A(s_i)} q_\pi(s_i, a)\pi(a|s_i)$$

$$= \sum_{a \in A(s_i)} \left(\rho(s_i, a) + \gamma \sum_{j=1}^{k} p_{a,ij} v_\pi(s_j) \right) \pi(a|s_i)$$

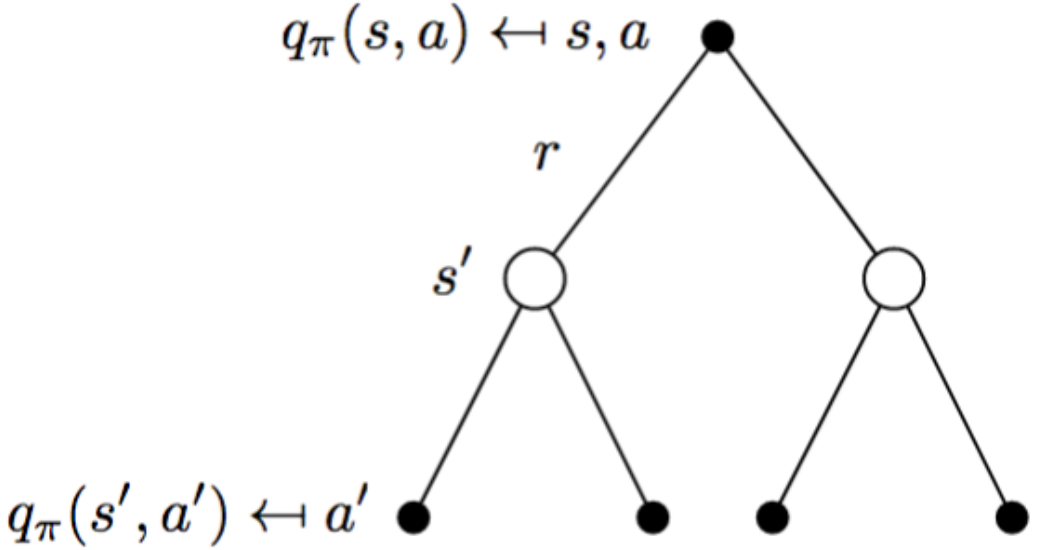


Bellman Equation for Action-Value Function

Recurrent expression for action-value function:

$$q_\pi(s_i, a) = \rho(s_i, a) + \gamma \sum_{j=1}^{k} p_{a,ij} v_\pi(s_j)$$

$$= \rho(s_i, a) + \gamma \sum_{j=1}^{k} p_{a,ij} \left(\sum_{a' \in A(s_j)} q_\pi(s_j, a') \pi(a'|s_j) \right)$$



MDP with Policy. Transition Probabilities

As soon as a policy π is applied to MDP, it becomes an MRP:

$$\text{MDP} + \text{policy} = \text{MRP}$$

For MRP:

$$p_{ij} = P(X_n = s_j \mid X_{n-1} = s_i), \quad i, j = 1, ..., k$$

For MDP:

$$p_{a,ij} = P(X_n = s_j \mid X_{n-1} = s_i, A_n = a), \quad i, j = 1, ..., k, \ a \in A(s_i)$$

For MDP with policy π:

$$p_{\pi,ij} = \sum_{a \in A(s_i)} p_{a,ij} \pi(a|s_i), \quad i, j = 1, ..., k$$

(by the law of total probability)

The only one probability transition matrix P_π is needed

MDP with Policy. Reward Function

For MRP:

$$\rho(s) = \mathrm{M}[R_{n+1} \mid X_n = s], \quad s \in S$$

For MDP:

$$\rho(s, a) = \mathrm{M}[R_{n+1} \mid X_n = s, A_n = a], \quad a \in A(s),\ s \in S$$

For MDP with policy π:

$$\rho_\pi(s) = \mathrm{M}[\rho(s, a(s))] = \sum_{a \in A(s)} \rho(s, a)\pi(a|s), \quad s \in S$$

where $a(s)$ is action taken in state s w.r.t. policy π (deterministic or random)

The reward function $\rho_\pi(s)$ is the function of states only:

$$\rho_\pi : S \to \mathbb{R}$$

MDP with Policy. Bellman Equation

For MRP:

$$v(s_i) = \rho(s_i) + \gamma \sum_{j=1}^{k} p_{ij} v(s_j), \quad i = 1, ..., k$$

For MDP with policy π:

$$\begin{aligned} v_\pi(s_i) &= \sum_{a \in A(s_i)} \left(\rho(s_i, a) + \gamma \sum_{i=1}^{k} p_{a,ij} v_\pi(s_j) \right) \pi(a|s_i) \\ &= \sum_{a \in A(s_i)} \rho(s_i, a) \pi(a|s_i) + \gamma \sum_{j=1}^{k} \left(\sum_{a \in A(s_i)} p_{a,ij} \pi(a|s_i) \right) v_\pi(s_j) \\ &= \rho_\pi(s_i) + \gamma \sum_{j=1}^{k} p_{\pi,ij} v_\pi(s_j), \quad i = 1, ..., k \end{aligned}$$

MDP with Policy. Solution of Bellman Equation

The value function $v_\pi(s)$ of MDP with given policy π can be found directly, it is the solution of the Bellman equation for the corresponding MRP

Bellman equation in matrix form:

$$v_\pi = \rho_\pi + \gamma P_\pi v_\pi$$

$$\begin{pmatrix} v_\pi(s_1) \\ \dots \\ v_\pi(s_k) \end{pmatrix} = \begin{pmatrix} \rho_\pi(s_1) \\ \dots \\ \rho_\pi(s_k) \end{pmatrix} + \gamma \begin{pmatrix} p_{\pi,11} & \dots & p_{\pi,1k} \\ \dots & \dots & \dots \\ p_{\pi,k1} & \dots & p_{\pi,kk} \end{pmatrix} \begin{pmatrix} v_\pi(s_1) \\ \dots \\ v_\pi(s_k) \end{pmatrix}$$

Direct solution:

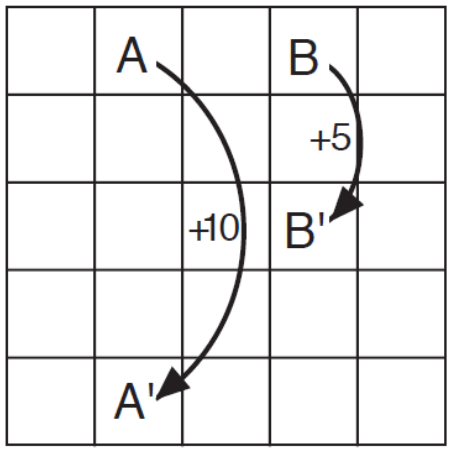
$$v_\pi = (I - \gamma P_\pi)^{-1} \rho_\pi$$

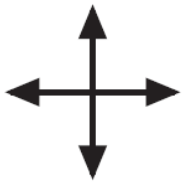
In practice, it can be solved only for small number of states k, because of the inverse operation (complexity $O(k^3)$)

Iterative algorithms can be applied to solve it

Gridworld Example. Description

Consider an MDP with states that correspond to the cells of a gridworld. At each cell four actions are possible: north, south, east and west, which deterministically cause the chain to move one cell in the respective direction on the grid




Actions

$k = |S| = 5 \cdot 5 = 25$,
$A(s) = \{\text{north, south, east, west}\}$ for all $s \in S$
All transition probabilities $p_{a,ij}$ are equal to 1 or 0 for all $s_i \in S$, $s_j \in S$, $a \in A(s_i)$, $i, j = 1, ..., 25$

Gridworld Example. Rewards

- Actions that would take out of the grid leave its location unchanged, but also result in a reward of –1
- Other actions result in a reward of 0, except those that move the chain out of the special states A and B
- From state A, all four actions move the chain to state A' and yield a reward of +10
- From state B, all four actions move the chain to state B' and yield a reward of +5

Formally:

$R(A, a, A') = 10$, $R(B, a, B') = 5$ for all actions a

$R(s, a, s) = -1$ for border states s and actions a that would take out of the grid

$R(s, a, s') = 0$ for all other triples (s, a, s')

Gridworld Example. Reward Function

Suppose the decision maker applies the following policy π:
randomly and equally probable choose an action from set $A(s) = \{\text{north, south, east, west}\}$ in all states $s \in S$

The reward function of state $s \in S$ of gridworld MDP with policy π is the expected reward received upon leaving state s:

$$\rho_\pi(s) = \sum_{a \in A(s)} \rho(s,a)\pi(a|s) = 0.25 \sum_{a \in A(s)} \rho(s,a)$$

(by the law of total probability)

The reward function of state-action pair (s,a):

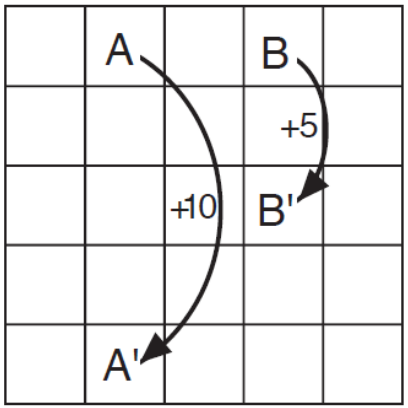
$$\rho(s,a) = \mathrm{M}[R_{n+1} \mid X_n = s, A_n = a]$$

(doesn't depend on policy π)

Gridworld Example. Reward Function

Since the successor state s' is deterministic given the current state s and action a taken, reward function $\rho(s, a)$ is deterministic for all state-action pairs (s, a) and equal to 0, –1, +10 or +5

$$\rho_\pi(s_{11}) = 0.25(-1 + 0 + 0 + (-1)) = -0.5$$
$$\rho_\pi(A) = 0.25(10 + 10 + 10 + 10) = 10$$
$$\rho_\pi(B) = 0.25(5 + 5 + 5 + 5) = 5$$
$$\rho_\pi(A') = 0.25(0 + (-1) + 0 + 0) = -0.25$$
$$\rho_\pi(B') = 0.25(0 + 0 + 0 + 0) = 0$$



Gridworld

−0.5	10.0	−.25	5.0	−0.5
−.25	0	0	0	−.25
−.25	0	0	0	−.25
−.25	0	0	0	−.25
−0.5	−.25	−.25	−.25	−0.5

ρ_π

Gridworld Example. State Transition Probabilities

State transition probabilities of gridworld MDP with policy π:

$$p_{\pi,ij} = \sum_{a \in A(s_i)} p_{a,ij} \pi(a|s_i) = 0.25 \sum_{a \in A(s_i)} p_{a,ij}, \quad i,j = 1,...,25$$

State transition probabilities for action a:

$$p_{a,ij} = P(X_{n+1} = s_j \mid X_n = s_i, A_n = a), \quad i,j = 1,...,25$$

There are four 25×25 transition probability matrices:
P_{north}, P_{south}, P_{east}, P_{west}

Probabilities for transition $s_{11} \to s_{11}$:

$p_{\text{north},11,11} = 1$, $p_{\text{south},11,11} = 0$, $p_{\text{east},11,11} = 0$, $p_{\text{west},11,11} = 1$
$p_{\pi,11,11} = 0.25(1 + 0 + 0 + 1) = 0.5$

Probabilities for transition $s_{11} \to s_{12}$:

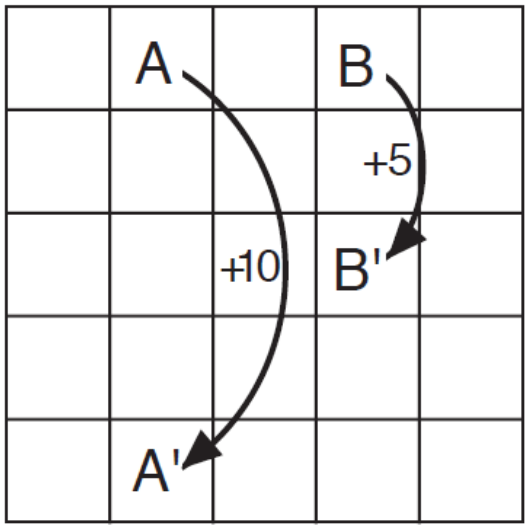
$p_{\text{north},11,12} = 0$, $p_{\text{south},11,12} = 0$, $p_{\text{east},11,12} = 1$, $p_{\text{west},11,12} = 0$
$p_{\pi,11,12} = 0.25(0 + 0 + 1 + 0) = 0.25$

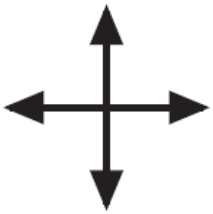
Gridworld Example. State-Value Function

The state-value function of gridworld MDP with policy π is the solution of the Bellman equation:

$$v_\pi = (I - \gamma P_\pi)^{-1} \rho_\pi$$

The state-value function for $\gamma = 0.9$





Actions

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

Gridworld Example. Notes

- The negative values near the lower edge
  These are the result of the high probability of hitting the edge of the grid there under the random policy π
- State A is the best state to be in under this policy, but its expected return (+8.8) is less than immediate reward (+10)
  From A the chain is taken to A', from which it is likely to run into the edge of the grid
- State B, on the other hand, is valued by +5.3 that is more than its immediate reward +5
  From B the chain is taken to B', which has a positive value (+0.4). From B' the expected penalty (negative reward) for possibly running into an edge is more than compensated by the expected gain for possibly stumbling onto A or B