

Markov Decision Processes

- The Markov Property
- The Markov Decision Process
- Partially Observable MDPs

The Premise

Much of the time, statistics are thought of as being very *deterministic*, for example:

79.8% of Stanford students graduate in 4 years.

(www.collegeresults.org, 2014)

It's very tempting to read this sort of statistic as if *graduating from Stanford in four years* is a randomly determined event. In fact, it's a combination of two things:

- random chance, which does play a role
- the actions of the student

The Premise

Situations like this, in which outcomes are a combination of random chance and active choices, can still be analyzed and optimized.

You can't optimize the part that happens **randomly**, but you can still find the best **actions** to take: in other words, you can optimize your actions to find the best outcome in spite of the random element.

One way to do this is by using a **Markov Decision Process**.

The Markov Property

Markov Decision Processes (MDPs) are **stochastic processes** that exhibit the **Markov Property**.

- Recall that **stochastic processes**, in unit 2, were processes that involve randomness. The examples in unit 2 were not influenced by any active choices – everything was random. This is why they could be analyzed without using MDPs.
- The **Markov Property** is used to refer to situations where the probabilities of different outcomes are not dependent on past states: the current state is all you need to know. This property is sometimes called “memorylessness”.

The Markov Property

For example, if an unmanned aircraft is trying to remain level, all it needs to know is its **current state**, which might include how level it currently is, and what influences (momentum, wind, gravity) are acting upon its state of level-ness. This analysis displays the Markov Property.

In contrast, if an unmanned aircraft is trying to figure out how long until its battery dies, it would need to know not only how much charge it has now, but also how fast that charge has declined from a **past state**. Therefore, this does not display the Markov Property.

Practice Problem 1

1. For each scenario:

- tell whether the outcome is influenced by chance only or a combination of chance and action
 - tell whether the outcomes' probabilities depend only on the present or partially on the past as well
 - argue whether this scenario can or can not be analyzed with an MDP
- a. A Stanford freshman wants to graduate in 4 years.
 - b. An unmanned vehicle wants to avoid collision.
 - c. A gambler wants to win at roulette.
 - d. A truck driver wants to get from his current location to Los Angeles.

Aspects of an MDP

Some important aspects of a Markov Decision Process:

State: a set of existing or theoretical conditions, like position, color, velocity, environment, amount of resources, etc.

One of the challenges in designing an MDP is to figure out what all the possible states are. The current state is only one of a large set of possible states, some more desirable than others.

Even though an object only has one current state, it will probably end up in a different state at some point.

Aspects of an MDP

Action: just like there is a large set of possible states, there is also a large set of possible actions that might be taken.

The current state often influences which actions are available. For example, if you are driving a car, your options for *turning left or right* are often restricted by what lane you are in.

Aspects of an MDP

A **Probability Distribution** is used to determine the transition from the current state to the next state.

The probability of one state (flying sideways) leading to another (crashing horribly) is influenced by both action and chance. The same state (flying sideways) may also lead to other states (recovering, turning, landing safely) with different probabilities, which also depend on both actions and chance.

All of these different probabilities are called the *probability distribution*, which is often contained in a matrix.

Aspects of an MDP

The last aspect of an MDP is an artificially generated **reward**. This reward is calculated based on the value of the next state compared to the current state. More favorable states generate better rewards.

For example, if a plane is *flying sideways*, the reward for *recovering* would be much higher than the reward for *crashing horribly*.

The Markov Decision Process

Once the states, actions, probability distribution, and rewards have been determined, the last task is to run the process. A **time step** is determined and the state is monitored at each time step.

In a simulation,

1. the initial state is chosen randomly from the set of possible states.
2. Based on that state, an action is chosen.
3. The next state is determined based on the probability distribution for the given state and the action chosen.
4. A reward is granted for the next state.
5. The entire process is repeated from step 2.

Practice Problem 2

2. Suppose a system has two states, **A** (good) and **B** (bad). There are also two actions, **x** and **y**.

From state A, **x** leads to A (60%) or B (40%); **y** leads to A (50%) or B (50%).

From state B, **x** leads to A (30%) or B(70%); **y** leads to A(80%) or B(20%).

- a) Run a simulation starting in state A and repeating action x ten times. Use `rand()` to generate the probabilities. For each time step you are in state A, give yourself 1 point.
- b) Run the same simulation using action y ten times.
- c) Run the same simulation alternating x and y.

Practice Problem 3

Write two functions in Julia, `actionx(k)` and `actiony(k)`, that take an input `k` (the state A or B) and return an output “A” or “B” based on the probabilities from the preceding problem:

From state A, x leads to A (60%) or B (40%); y leads to A (50%) or B (50%). From state B, x leads to A (30%) or B(70%); y leads to A(80%) or B(20%).

Use your program in a simulation to find the total reward value of repeating `x` 100 times or repeating `y` 100 times. Use multiple simulations. Which is better?

Probability Distribution Matrices

In the previous problem, the probabilities for action x were as follows: *From state A, x leads to A (60%) or B (40%); from state B, x leads to A (30%) or B(70%).*

This information can be summarized more briefly in a matrix:

	A	B
A	.6	.4
B	.3	.7

current state

next state

Matrices enable neat probability summaries even for very large sets of possible events or “event spaces”.

Practice Problem 4

Given this probability distribution for action k:

	A	B	C
A	.1	.2	.7
B	.8	0	.2
C	0	.4	.6

tell the probability that action k will move from state...

a) A to B

b) A to C

c) C to A

d) C to C

e) B to C

f) B to A

g) Why do the rows have to add up to 1, but the columns not?

Intelligent Systems

In practice problem 2, the action taken did not depend on the state: x was chosen or y was chosen regardless of the current state.

However, given that A was the desired state, the **most intelligent** course of action would be the one most likely to return A, which was a different depending on whether the current state was A or B.

A system that monitors its own state and chooses the best action based on its state is said to display **intelligence**.

Practice Problem 5

Here are the matrices for actions x and y again:

action x: $\begin{bmatrix} .6 & .4 \\ .3 & .7 \end{bmatrix}$, action y: $\begin{bmatrix} .5 & .5 \\ .8 & .2 \end{bmatrix}$

- In state A, what action is the best?
- In state B, what action is the best?
- Write a program in Julia that chooses the best action based on the current state. Run several 100-time-step simulations using your intelligent program.

Applications of MDPs

Most systems, of course, are far more complex than the two-state, two-action example given here. As you can imagine, running simulations for every possible action in every possible state requires some powerful computing.

MDPs are used extensively in robotics, automated systems, manufacturing, and economics, among other fields.

POMDPs

A variation on the traditional MDP is a **Partially Observable Markov Decision Process** (POMDP, pronounced “Pom D.P.”). In these scenarios, the system does not know exactly what state it is currently in, and therefore has to guess.

This is like the difference between thinking,

“I’m going in the right direction”

and

“I **think** I’m going in the right direction”.

POMDPs

POMDPs have the same elements as a traditional MDP, plus two more.

The first is the **belief state**, which is the state the system believes it is in. The belief state is a probability distribution.

For example,

“I think I’m going in the right direction”

might really mean:

- 80% chance this is the right direction
- 15% mostly-right direction
- 5% completely wrong direction

} probability
distribution!



POMDPs

The second additional element is a set of **observations**. After the system takes an action based on its belief state, it observes what happens next and updates its belief state accordingly.

For example, if you took a right turn and didn't see the freeway you expected, you would then change your probability distribution for whether you were going in the right direction.

POMDPs

Otherwise, a POMDP works much like a traditional MDP:

- An action is chosen based on the **belief state**  traditional: current state
- The next state is reached and the reward granted
- **an observation is made and the belief state updated.**  traditional: this step absent

Then the next action is chosen based on the belief state and the process repeats.

Applications of POMDPs

POMDPs are used in robotics, automated systems, medicine, machine maintenance, biology, networks, linguistics, and many other fields.

There are also many variations on POMDP frameworks depending on the application. Some rely on Monte Carlo-style simulations. Others are used in machine learning applications (in which computers learn from trial and error rather than relying only on programmers' instructions).

Because POMDPs are widely applicable and also fairly new on the scene, fresh variations are in constant development.

Practice Problem 6

Imagine that you are a doctor trying to help a sneezy patient.

- a. Name three possible states for “Why this patient sneezes” and assign a probability to each state. (This is your belief state.)
- b. Choose an action based on your belief state.
- c. Suppose the patient is still sneezing after your action. Update your belief state.
- d. Write a paragraph explaining your reasoning in a – c, and what other steps you might take (including questions to ask or other actions to try) to refine your belief state.