# Hidden Markov Models and Reinforcement Learning

## Contents

- Concept and examples of Markov models
- Towards Hidden Markov models (HMM)
- Applications of HMM in Language models
- Concept of Reinforcement Learning

Let’s begin with Markov Models….

## Definition

- A Markov Model is a stochastic model which models temporal or sequential data, i.e., data that are ordered.
- It provides a way to model the dependencies of current information (e.g. weather) with previous information.
- It is composed of states, transition scheme between states,and emission of outputs (discrete or continuous)
- Autocomplete / Next word prediction

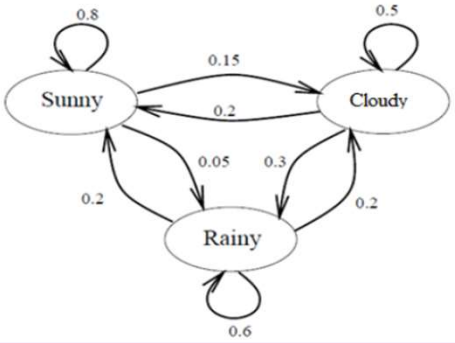
## Hidden Markov Models

## Hidden Markov Models

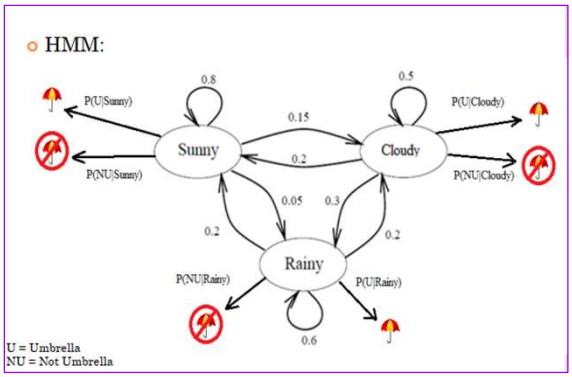
- A Hidden Markov Model, is a stochastic model where the states of the model are hidden. Each state can emit an output which is observed.
- A scenario : Imagine you were locked in a room for several days and you were asked about the weather outside. The only piece of evidence you have is whether the person
- who comes into the room bringing your daily meal is carrying an umbrella or not.
- What is hidden? Sunny, Rainy, Cloudy
- What can you observe? Umbrella or Not

## Markov Model vs Hidden Markov Model

- Markov Chain:




- HMM:




- Let's assume that $t$ days had passed. Therefore, we will have an observation sequence $O = \{o_1, \dots, o_t\}$, where $o_i \epsilon \{Umbrella, Not\ Umbrella\}$.
- Each observation comes from an unknown state. Therefore, we will also have an unknown sequence $Q = \{q_1, \dots, q_t\}$, where $q_i \epsilon \{Sunny, Rainy, Cloudy\}$.
- We would like to know: $P(q_1, .., q_t | o_1, \dots, o_t)$.

## HMM Mathematical Model

- From Bayes' Theorem, we can obtain the probability for a particular day as:

$$P(q_i|o_i) = \frac{P(o_i|q_i)P(q_i)}{P(o_i)}$$

For a sequence of length $t$:

$$P(q_1, \dots, q_t|o_1, \dots, o_t) = \frac{P(o_1, \dots, o_t|q_1, \dots, q_t)P(q_1, \dots, q_t)}{P(o_1, \dots, o_t)}$$

- From the Markov property:

$$P(q_1, \dots, q_t) = \prod_{i=1}^{t} P(q_i|q_{i-1})$$

- Independent observations assumption:

$$P(o_1, \dots, o_t|q_1, \dots, q_t) = \prod_{i=1}^{t} P(o_i|q_i)$$

$$P(q_1, \dots, q_t|o_1, \dots, o_t) \propto \underbrace{\prod_{i=1}^{t} P(o_i|q_i) \prod_{i=1}^{t} P(q_i|q_{i-1})}$$

**HMM Parameters:**

- Transition probabilities $P(q_i|q_{i-1})$
- Emission probabilities $P(o_i|q_i)$
- Initial state probabilities $P(q_i)$

## HMM Parameters

- A HMM is governed by the following parameters:

$$\lambda = \{A, B, \pi\}$$

  - State-transition probability matrix $A$
  - Emission/Observation/State Conditional Output probabilities $B$
  - Initial (prior) state probabilities $\pi$

- Determine the fixed number of states ($N$):

$$S = \{s_1, \ldots, s_N\}$$

- State-transition probability matrix:

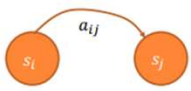
$$A = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1N} \\ a_{21} & a_{23} & \cdot & \cdot & \cdot & a_{2N} \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ a_{N1} & a_{N2} & \cdot & \cdot & \cdot & a_{NN} \end{bmatrix}$$

$\sum_{j=1}^{N} a_{ij} = 1$ (Each row/Outgoing arrows)

$a_{ij} = P(q_t = s_j \mid q_{t-1} = s_i), \quad 1 \le i, j \le N$

$a_{ij} \ge 0$

$a_{ij} \rightarrow$ *Transisiton probability from state* $s_i$ *to* $s_j$

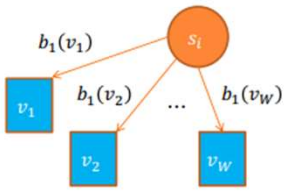



- Emission probabilities: A state will generate an observation (output), but a decision must be taken according on how to model the output, i.e., as discrete or continuous.
  - **Discrete outputs are modeled using pmfs.**
  - Continuous outputs are modeled using pdfs.

- Discrete Emission Probabilities:

Observation Set: $V = \{v_1, \ldots, v_W\}$

$b_i(v_k) = P(o_t = v_k | q_t = s_i), \quad 1 \le k \le W$




$$B = \begin{bmatrix} b_1(v_1)\, b_1(v_2) & \cdot & \cdot & \cdot & b_1(v_W) \\ b_2(v_1)\, b_2(v_2) & \cdot & \cdot & \cdot & b_2(v_W) \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ b_N(v_1) b_N(v_2) & \cdot & \cdot & \cdot & b_N(v_W) \end{bmatrix}$$

- Initial (prior) probabilities: these are the probabilities of starting the observation sequence in state $q_i$.

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \cdot \\ \cdot \\ \cdot \\ \pi_N \end{bmatrix} \qquad \pi_i = P(q_1 = s_i), \quad 1 \le i \le N$$

$$\sum_{i=1}^{N} \pi_i = 1$$

# HMM Example: Coins & Dice

$P(H|Red\ Coin) = 0.9$

$P(T|Red\ Coin) = 0.1$

$Outputs = \{1, 2, 3, 4, 5, 6\}$

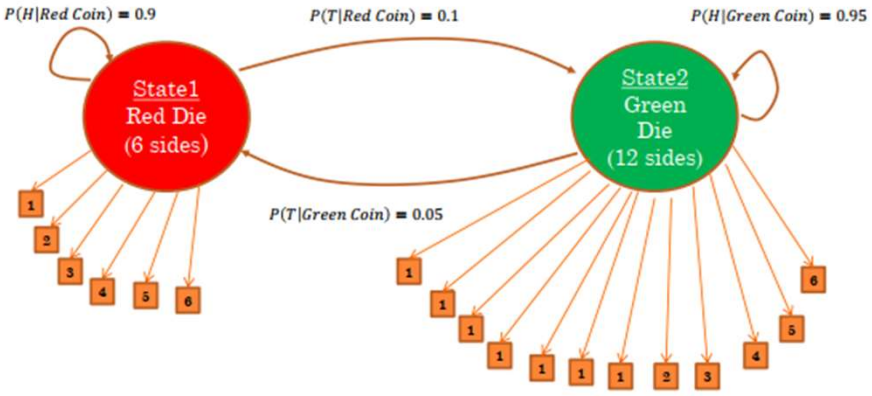
$P(H|Green\ Coin) = 0.95$

$P(T|Green\ Coin) = 0.05$


$Outputs = \{1, 1, 1, 1, 1, 1, 1, 2, 3, 4, 5, 6\}$

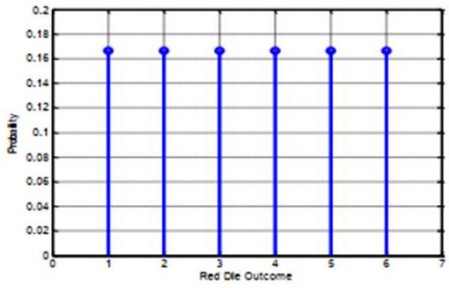
http://www.mathworks.com/help/stats/hidden-markov-models-hmm.html

## HMM Example: Coins & Dice

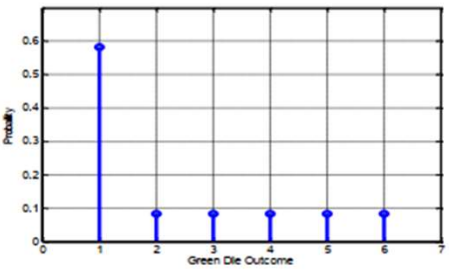



$$A = \begin{bmatrix} 0.9 & 0.1 \\ 0.05 & 0.95 \end{bmatrix} \qquad \pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

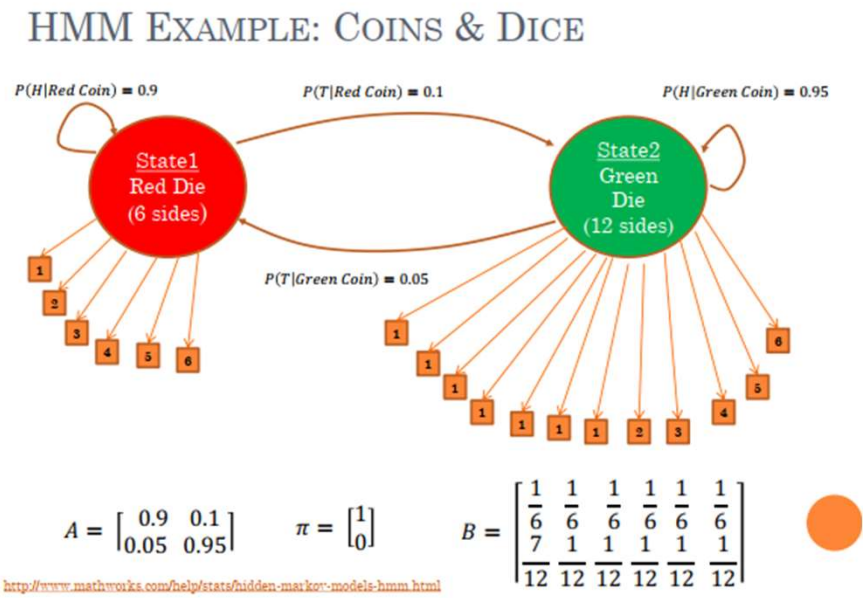
## HMM Example: Coins & Dice




$$b_1(o_t) = \{\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\}$$




$$b_2(o_t) = \{\frac{7}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}\}$$




## What next in HMM?

- Baum Welch algorithm : aims to tune the parameters of the HMM
- Viterbi algorithm : Given the optimized parameters, this algorithm predicts information about hidden states.

## Reinforcement Learning

## Reinforcement Learning

- Reinforcement Learning is an aspect of Machine learning where an agent learns to behave in an environment, by performing certain actions and observing the rewards/results which it get from those actions.

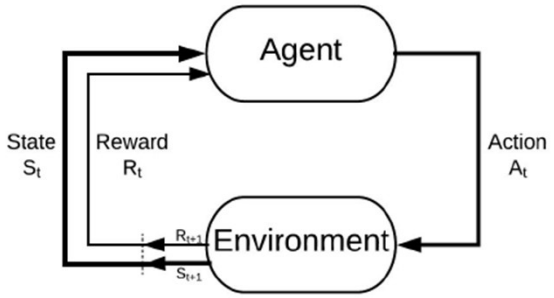
- Let’s understand with an example...

## Some terminologies

- **Agent:** It is an assumed entity which performs actions in an environment to gain some reward.
- **Environment (e):** A scenario that an agent has to face.
- **Reward (R):** An immediate return given to an agent when he or she performs specific action or task.
- **State (s):** State refers to the current situation returned by the environment.
- **Policy (π):** It is a strategy which applies by the agent to decide the next action based on the current state.
- **Value (V):** It is expected long-term return with discount, as compared to the short-term reward.
- **Value Function:** It specifies the value of a state that is the total amount of reward. It is an agent which should be expected beginning from that state.
- **Model of the environment:** This mimics the behavior of the environment. It helps you to make inferences to be made and also determine how the environment will behave.
- **Model based methods:** It is a method for solving reinforcement learning problems which use model-based methods.
- **Q value or action value (Q):** Q value is quite similar to value. The only difference between the two is that it takes an additional parameter as a current action.

## Principle of Reinforcement Learning (RL)

The objective of RL is to maximize the reward of an agent by taking a series of actions in response to a dynamic environment.

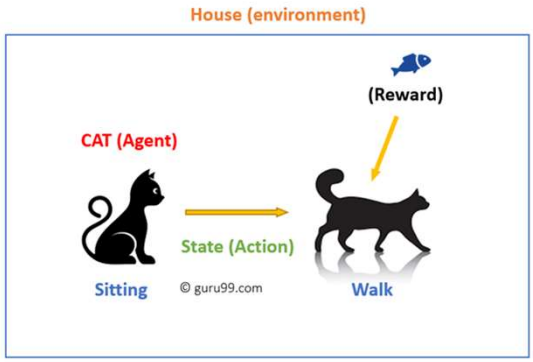


## Steps for Reinforcement Learning

1. Observation of the environment
2. Deciding how to act using some strategy
3. Acting accordingly
4. Receiving a reward or penalty
5. Learning from the experiences and refining our strategy
6. Iterate until an optimal strategy is found

## A naive Example

- As cat doesn't understand English or any other human language, we can't tell her directly what to do. Instead, we follow a different strategy.
- We emulate a situation, and the cat tries to respond in many different ways. If the cat's response is the desired way, we will give her fish.
- Now whenever the cat is exposed to the same situation, the cat executes a similar action with even more enthusiastically in expectation of getting more reward(food).
- That's like learning that cat gets from "what to do" from positive experiences.
- At the same time, the cat also learns what not do when faced with negative experiences.



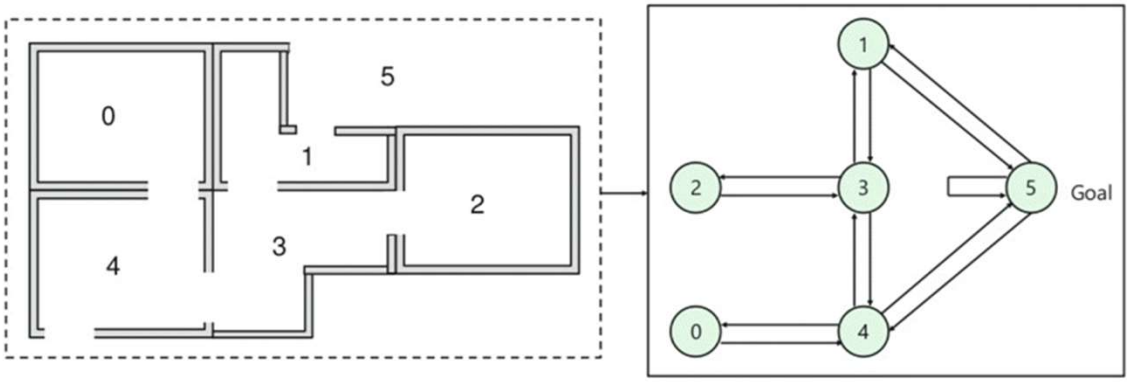

- Your cat is an agent that is exposed to the environment. In this case, it is your house. An example of a state could be your cat sitting, and you use a specific word in for cat to walk.
- Our agent reacts by performing an action transition from one "state" to another "state."
- For example, your cat goes from sitting to walking.
- The reaction of an agent is an action, and the policy is a method of selecting an action given a state in expectation of better outcomes.
- After the transition, they may get a reward or penalty in return.

## Q Learning

Q learning is a value-based method of supplying information to inform which action an agent should take.

- There are five rooms in a building which are connected by doors.
- Each room is numbered 0 to 4
- The outside of the building can be one big outside area (5)
- Doors number 1 and 4 lead into the building from room 5

## Scenario



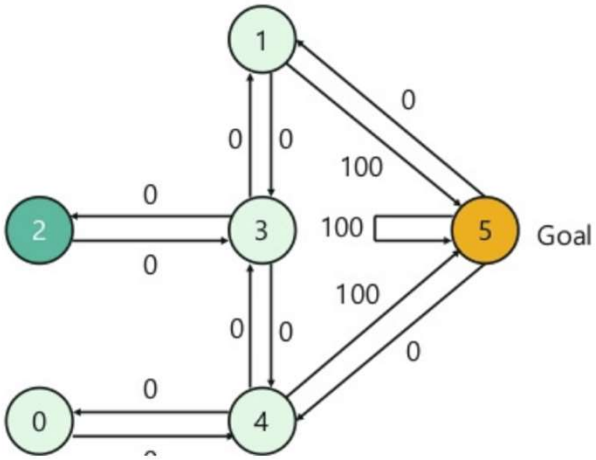

## Q Learning Example

Let us associate a reward value to each door:

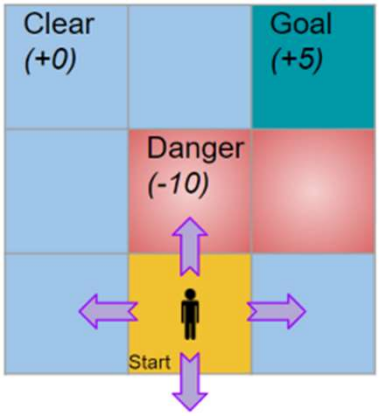
- Doors which lead directly to the goal have a reward of 100
- Doors which are not directly connected to the target room give zero reward
- As doors are two-way, and two arrows are assigned for each room
- Every arrow in the above image contains an instant reward value

## Scenario of the Example

- Room represents the state
- The movement of the agent represents the action
- A typical traversal sequence from room 2 to 5
  - Initial state = state 2
  - State 2-> state 3
  - State 3 -> state (2,1,4)
  - State 4-> state (0,5,3)
  - State 1-> state (5,3)
  - State 0-> state 4




## Another Example




- The player starts in the Start square and wants to reach the Goal square as their final destination, where they get a reward of 5 points.
- Some squares are Clear while some contain Danger, with rewards of 0 points and -10 points respectively.
- In any square, the player can take four possible actions to move Left, Right, Up, or Down.

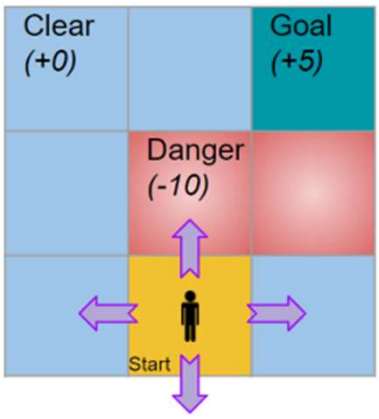
## The Q-Learning algorithm

- The Q-learning algorithm uses a Q-table of State-Action Values (also called Q-values).
- This Q-table has a row for each state and a column for each action.
- Each cell contains the estimated Q-value for the corresponding state-action pair.

## Process

- We start by initializing all the Q-values to zero.
- As the agent interacts with the environment and gets feedback, the algorithm iteratively improves these Q-values until they converge to the Optimal Q-values.
- It updates them using the Bellman equation.

## An Example




- This problem has 9 states since the player can be positioned in any of the 9 squares of the grid. It has 4 actions. So we construct a Q-table with 9 rows and 4 columns.

## An Example : Q Table

- Use the Q-table to lookup the Q-value for any state-action pair. eg. The value in a particular cell, say ((2, 2), Up) is the Q-value (or State-Action value) for the state (2, 2) and action 'Up'.
- All values are initialized to 0.

| | Left | Right | Up | Down |
|---|---|---|---|---|
| (1,1) | 0 | 0 | 0 | 0 |
| (1,2) | 0 | 0 | 0 | 0 |
| (1,3) | 0 | 0 | 0 | 0 |
| (2,1) | 0 | 0 | 0 | 0 |
| (2,2) | 0 | 0 | 0 | 0 |
| (2,3) | 0 | 0 | 0 | 0 |
| (3,1) | 0 | 0 | 0 | 0 |
| (3,2) | 0 | 0 | 0 | 0 |
| (3,3) | 0 | 0 | 0 | 0 |

## Q Learning

- Q-learning finds the Optimal policy by learning the optimal Q-values for each state-action pair.
- Initially, the agent randomly picks actions. But as the agent interacts with the environment, it learns which actions are better, based on rewards that it obtains. It uses this experience to incrementally update the Q values.

## Applications of Reinforcement Learning

- Robotics for industrial automation.
- Business strategy planning
- Machine learning and data processing
- It helps you to create training systems that provide custom instruction and materials according to the requirement of students.
- Aircraft control and robot motion control

## References

https://medium.com/analytics-vidhya/hidden-markov-model-part-1-of-the-hmm-series-3f7fea28a08

https://www.freecodecamp.org/news/a-brief-introduction-to-reinforcement-learning-7799af5840db/

https://towardsdatascience.com/reinforcement-learning-explained-visually-part-4-q-learning-step-by-step-b65efb731d3e

https://www.guru99.com/reinforcement-learning-tutorial.html