Hidden Markov Model(HMM)

- Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states.

- In probability theory, a Markov model is a stochastic model used to model randomly changing systems.

- It is assumed that future states depend only on the current state, not on the events that occurred before it (that is, it assumes the Markov property).

- It is used to predict future observations or classify sequences, based on the underlying hidden process that generates the data.

An HMM consists of two types of variables: hidden states and observations.

- The hidden states are the underlying variables that generate the observed data, but they are not directly observable.
- The observations are the variables that are measured and observed.

The Hidden Markov Model (HMM) is the relationship between the hidden states and the observations using two sets of probabilities: the transition probabilities and the emission probabilities.

- The transition probabilities describe the probability of transitioning from one hidden state to another.
- The emission probabilities describe the probability of observing an output given a hidden state.

HMM Algorithm

The Hidden Markov Model (HMM) algorithm can be implemented using the following steps:

- **Step 1: Define the state space and observation space**

The state space is the set of all possible hidden states, and the observation space is the set of all possible observations.

- **Step 2: Define the initial state distribution**

This is the probability distribution over the initial state.

- **Step 3: Define the state transition probabilities**

These are the probabilities of transitioning from one state to another. This forms the transition matrix, which describes the probability of moving from one state to another.

- **Step 4: Define the observation likelihoods:**

These are the probabilities of generating each observation from each state. This forms the emission matrix, which describes the probability of generating each observation from each state.

- **Step 5: Train the model**

The parameters of the state transition probabilities and the observation likelihoods are estimated using the Baum-Welch algorithm, or the forward-backward algorithm. This is done by iteratively updating the parameters until convergence.

- **Step 6: Decode the most likely sequence of hidden states**

Given the observed data, the Viterbi algorithm is used to compute the most likely sequence of hidden states. This can be used to predict future observations, classify sequences, or detect patterns in sequential data.

- **Step 7: Evaluate the model**

The performance of the HMM can be evaluated using various metrics, such as accuracy, precision, recall, or F1 score.

Example

A person Kia does either of these four things:

- Painting
- Cleaning the house
- Shopping for groceries
- Biking

Above are observables.

Assume that in four days Kia does the following: painting, cleaning, shopping, biking.

Hidden Markov Model consists of the following properties:

- **Hidden States S:** in the example above the hidden states are Sunny and Rainy, and they get grouped into a set S.

- **Observables O:** Paint, Clean, Shop and Bike. They get grouped into a set O.

- **Initial Probabilities** π: a matrix of the initial likelihood of the state at time t = 0.

In this case the likelihood that it is Sunny on the first day is 0.6, while the likelihood that it is Rainy is 0.4.

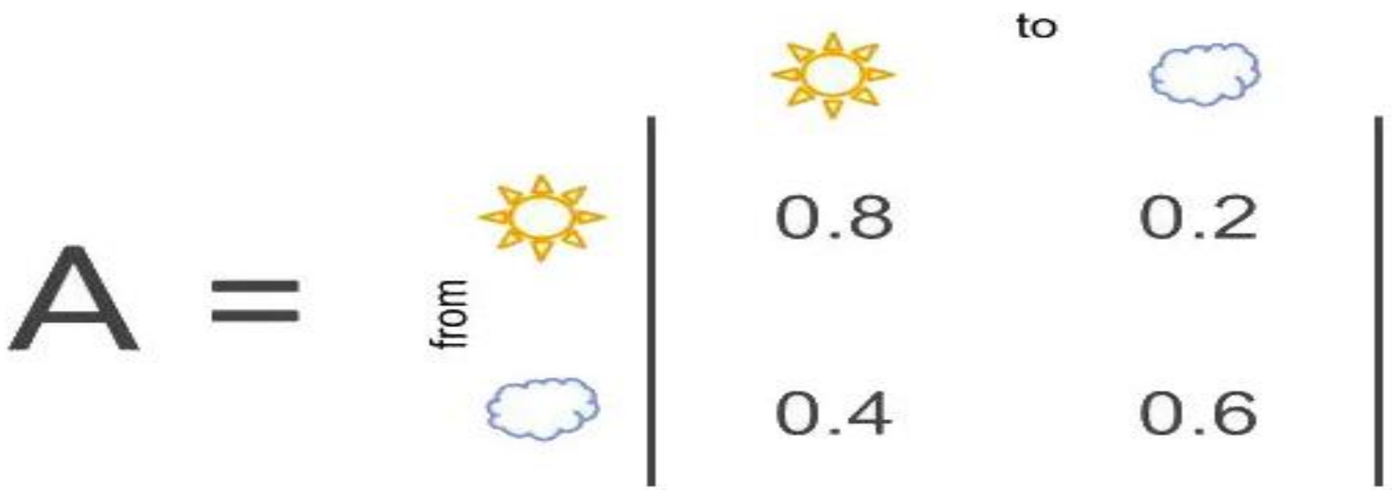
$\pi = |0.6, 0.4|$

Note: every row of the following matrices must add up to 1 since they represent a probability.

- **Transition Probabilities** A: a matrix that represents the probability of transitioning to another state given the current state.

- For example, if the current state is Sunny the probability that the day after is Sunny as well is 0.8, whereas the probability that the day after is Rainy is 0.2.

- Similarly if today is Rainy, the probability that tomorrow is Rainy as well is 0.6, while the probability that tomorrow is Sunny is 0.4.

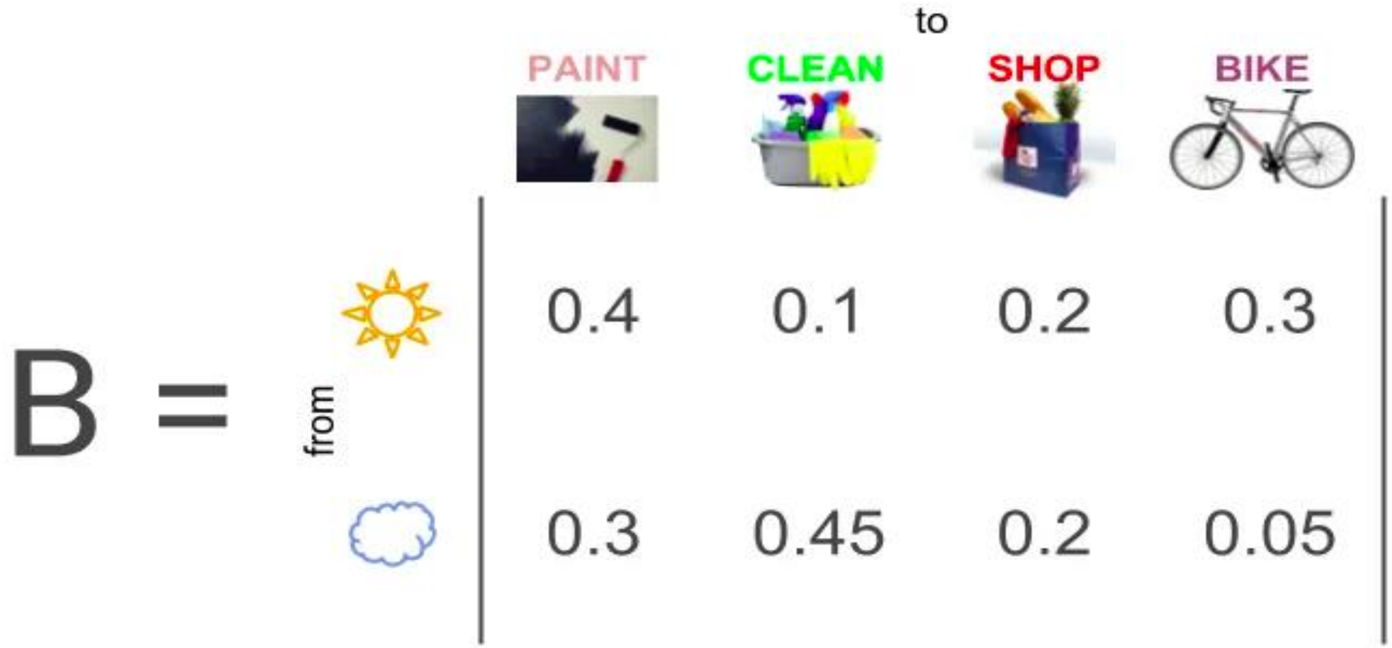


$$A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

Transition probabilities between states

- **Emission Probabilities B**: a matrix that represents the probability of seeing a specific observable given a hidden state.

For example, the probability of Clean on a Sunny day is 0.1, whereas the probability of Clean on a Rainy day is 0.45.



$$B = \begin{vmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{vmatrix}$$

Emission probabilities

Model Properties

$$S = \{ S_{sunny}, S_{rainy} \} \quad (Hidden\ States)$$

$$O = \{ O_{clean}, O_{bike}, O_{shop}, O_{paint} \} \quad (Observables)$$

$$\pi = \begin{vmatrix} 0.6 & 0.4 \end{vmatrix} \quad (Initial\ Probabilities)$$

$$A = \begin{vmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{vmatrix} \quad (Transition\ Probabilities)$$

$$B = \begin{vmatrix} 0.4 & 0.1 & 0.2 & 0.3 \\ 0.3 & 0.45 & 0.2 & 0.05 \end{vmatrix} \quad (Emission\ Probabilities)$$

Problem 1 — Likelihood

- Kia’s activity in four days. The observation sequence is as follows: Paint, Clean, Shop and Bike.

- So, what is the likelihood that this observation sequence O can derive from our HMM λ?
- There are two methods to calculate this: the Forward Algorithm and the Backward Algorithm.

- The Forward Algorithm
- The Forward algorithm comprises of three steps:

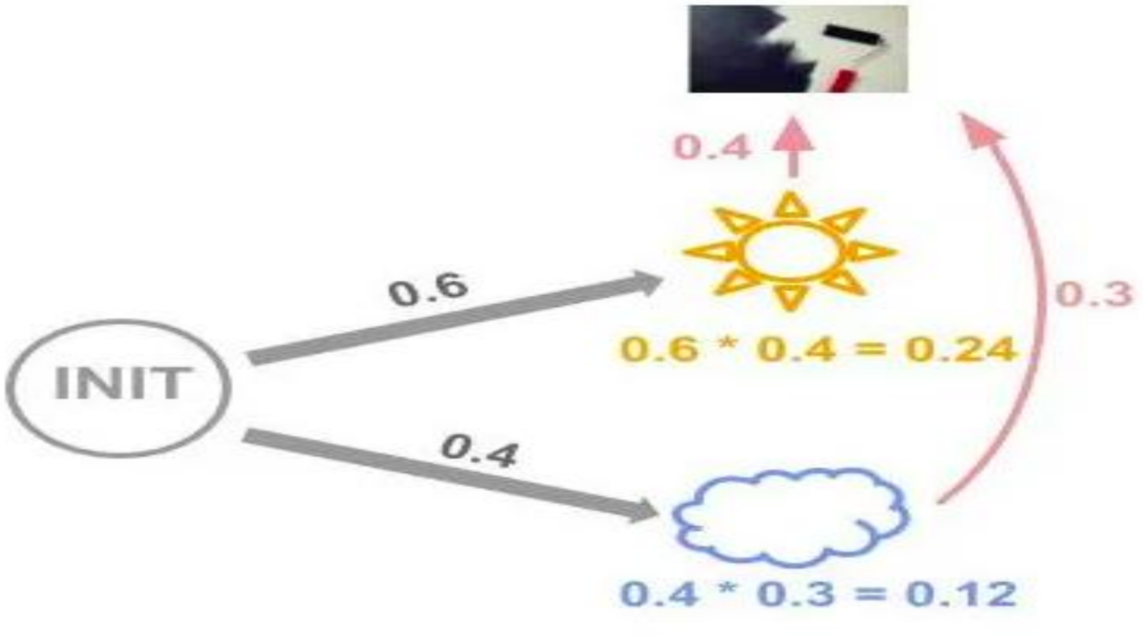
- Initialization
- Recursion
- Termination

- **Initialization**

$$\alpha_1(i) = \pi_i \cdot b_i(O_1)$$

Forward algorithm initialization equation

- The above equation means that the first forward variable is calculated by multiplying the initial probability of state i by the emission probability b of that state given the observable O at time 1.



Initialization of Forward Algorithm

- Initial forward variable of the Sunny state is the initial probability of Sunny, 0.6, times the emission probability from Sunny to the observable Paint, 0.4. Hence, 0.24.

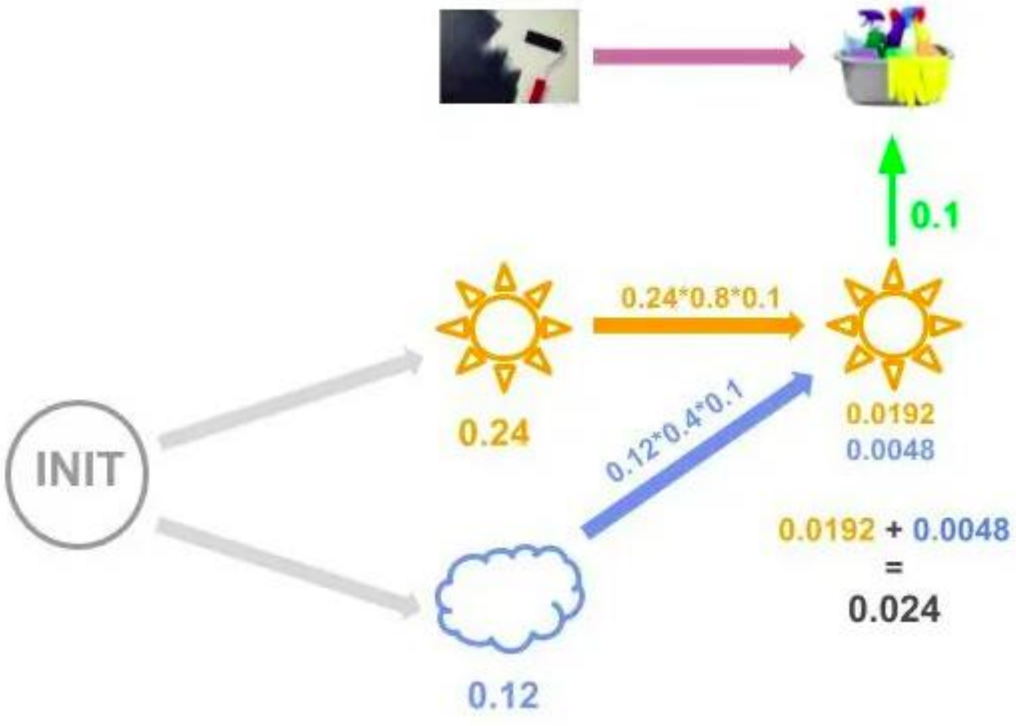
- While the initial forward variable of the Rainy state is the initial probability of Rainy, 0.4, times the emission probability from Rainy to the observable Paint, 0.3. Hence, 0.12.

Recursion

- For t = 1, 2, …, T-1 we make use of the recursion equation which defines the forward variable of state j as the product of the previous forward variable of state i, multiplied by the transition probability a between the previous state i to state j, multiplied by the emission probability b from state j to the observable O.

$$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_j(O_{t+1})$$

Forward algorithm recursion equation



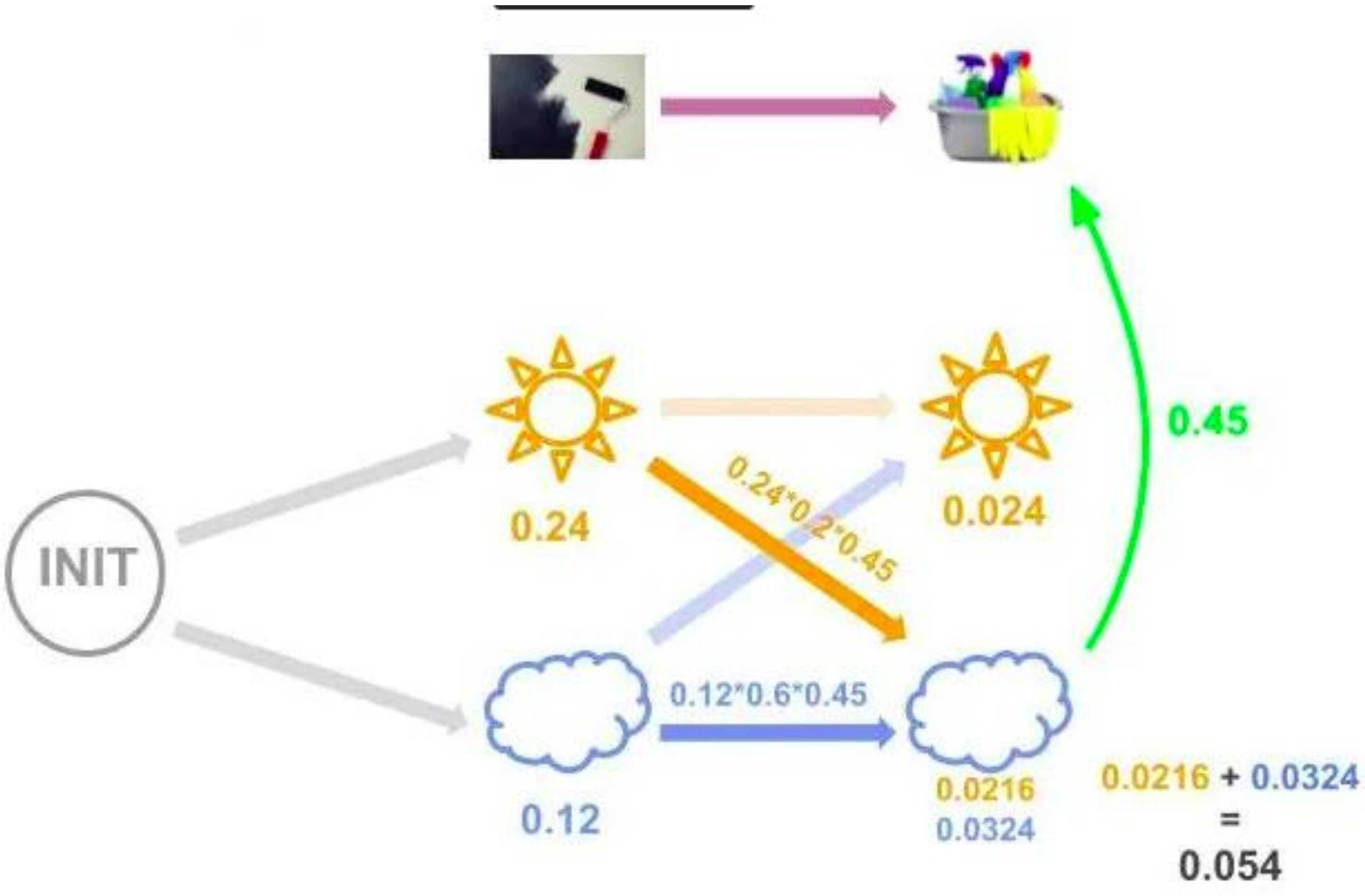
Recursion of Forward Algorithm 1

- Here we are calculating the forward variable of state Sunny at time 2 by summing the results of two multiplications:

- The previous forward variable of the previous Sunny state, 0.24, times the transition probability from Sunny to Sunny, 0.8, times the emission probability from Sunny to Clean, 0.1.
- 0.24 * 0.8 * 0.1 = 0.0192
- The previous forward variable of the previous Rainy state, 0.12, times the transition probability from Rainy to Sunny, 0.4, times the emission probability from Sunny to Clean, 0.1.
- 0.12 * 0.4 * 0.1 = 0.0048
- Then, according to the equation above, we sum these results and get our forward variable.

- α = 0.0192 +0.0048 = 0.024

- Similarly, for the next step forward variable of 0.054 for the Rainy state:



Recursion of Forward Algorithm 2

Termination

- will continue till reach the biking..........

- This final equation tells us that to find the probability of an observation sequence O deriving from an HMM model λ, need to sum up all the forward variables at time T, i.e. all the variables of every state at the end of the observation sequence.

Hence, in the example above after biking,

- $P(O|\lambda) = 0.0028512 + 0.0003048 = 0.003156$

- So, Kia has spent the last four days painting, cleaning, shopping and biking, is 0.003156 given the HMM model built for kia.

Hidden Markov Model applications

- Reinforcement learning and temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics.