Hidden Markov Models

Hidden Markov Models are probabilistic models

They are used to find probabilities of sequences of events.

Based on the sequences, questions can be answered

Day	Temperature	Icecreames eaten
Monday	Hot	2
Tuesday	Hot	1
Wednesday	Cold	0

What is the probability that I will eat 2 icecreames tomorrow?

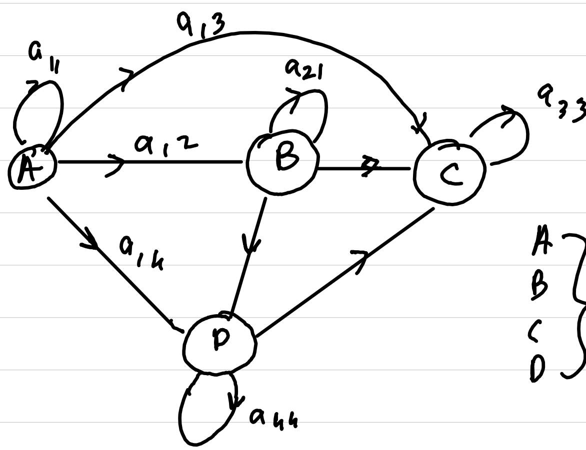
What is the temperature on thursday & friday if I ate 2 icecreames on Friday?

HMMs are kind of system called Finite or Discrete Markov model.

Markov model is a finite state machine with N distinct states begins at $t = 1$ in Initial state

Moves from one state to another state according to probabilities associated with current state

Number of states are finite



A
B
C
D
} states

a_{ij} is probability of moving from state a_i to a_j

$$\sum_{i=1}^{N} a_{ik} = 1 \quad \forall k$$

ie sum of all outgoing arrows = 1

Hidden Markov Model

HMM is a stastical model in which the system is assumed to be markov process with unobserved hidden states

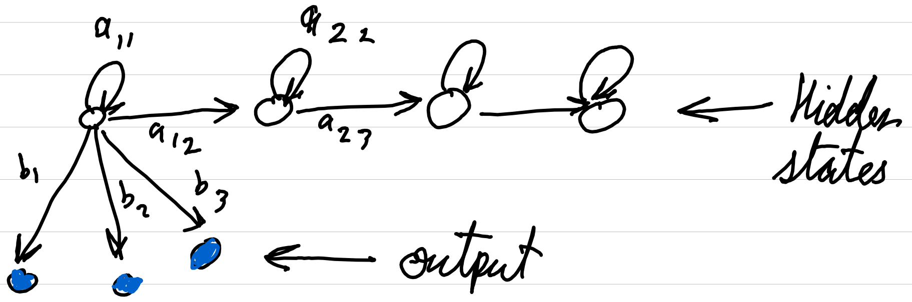
Consists of states $S_1, S_2, S_3 \ldots\ldots$

$$P(S_{ij} \mid S_{i_1}, S_{j_2}, \ldots S_{i_{k-1}}) = P(S_{i_k} \mid S_{i_{k-1}})$$
(Markov property)

$A \rightarrow$ set of transition probabilities

$B \rightarrow$ set of output probabilites

$\pi \rightarrow$ initial probabilities



$$\left.\begin{aligned} b_{11} + b_{12} + b_{13} + b_{14} &= 1 \\ b_{21} + b_{22} + b_{23} + b_{24} &= 1 \end{aligned}\right\} \text{output probability}$$

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

Markov property : The current state of system depends only on the previous state of system

state of time T_{+1} depends on state at T

Markov models obey the Markov property

Given $M(A, B, \pi)$ & a sequence O

We can do solve 3 problems →

O is a sequence (No of icecreames eaten this week
1, 2, 0, 0, 2, 2, 1)

States are hidden states traversed
(Temperature of week
Hot, Hot, Cold . . . . .)

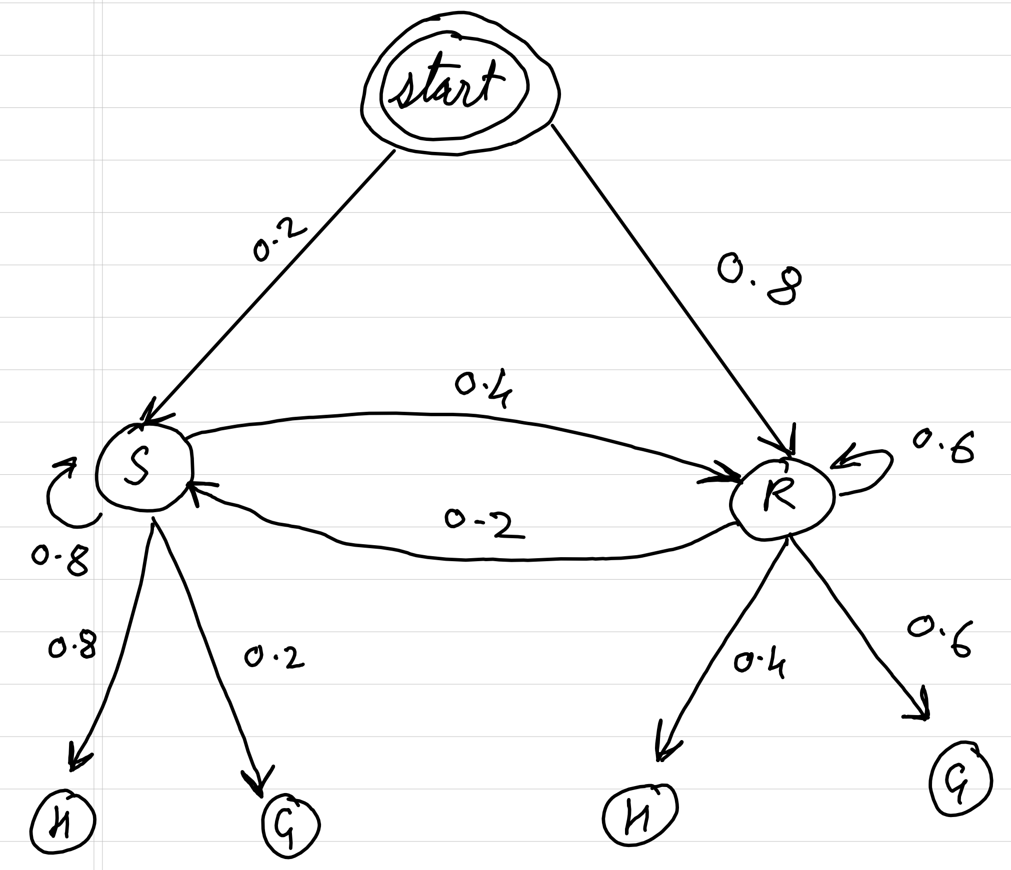
1. Evaluation problem → calculate the probability that model generates O

2. Decoding problem → calculate the most likely sequence of states visited for O

3. Learning problem → Determine HMM parameters that fit O

- HMM
  - Evaluation
  - Decoding
  - Learning
    - Supervised
      - Gradient
      - Viterbi
    - Unsupervised
      - Gradient
      - EM
      - Maximum Likelyhood

① Given Markov Model



S – Sunney Day   H – Happy
R – Rainy day   G – Grumpy

Given Emotion recorded on 6 days

Day	1	2	3	4	5	6
Emotion	H	H	G	G	G	H

Find the most likely climate on the 6 days

→ The Viterbi Algorithm is a dynamic programming algorithm.

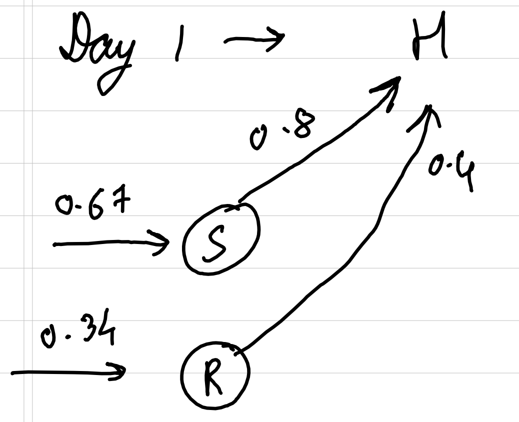
It tracks the maximum probability and corresponding state sequence.

There may be many paths that lead to the following emotions

eg every day might be sunny yet these emotions might be there.

But we need to find the most likely path

Day 1 →



S → 0.67 × 0.8 = 0.533

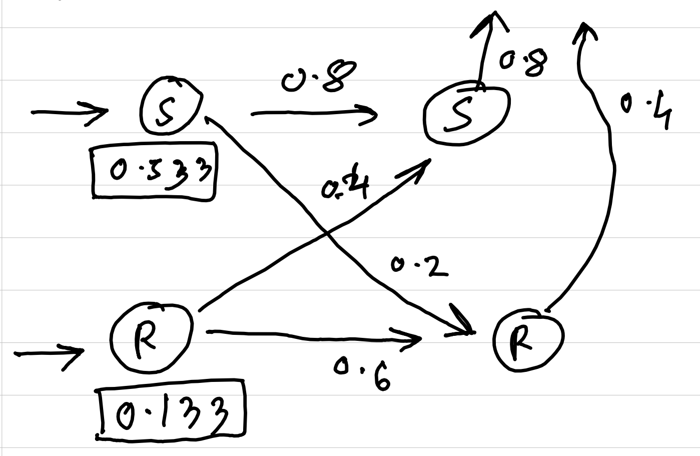
R → 0.34 × 0.4 = 0.133

Here the probability that the day will be sunny & emotion will be happy is 0.533

It is higher as opposed to day will be rainy and emotion will be happy

So we can say that Day 1 might be Sunny

Day 2   H



Sunny

0.533 × 0.8 × 0.8
= 0.341 ✓

0.133 × 0.4 × 0.8
= 0.00425

Rainy

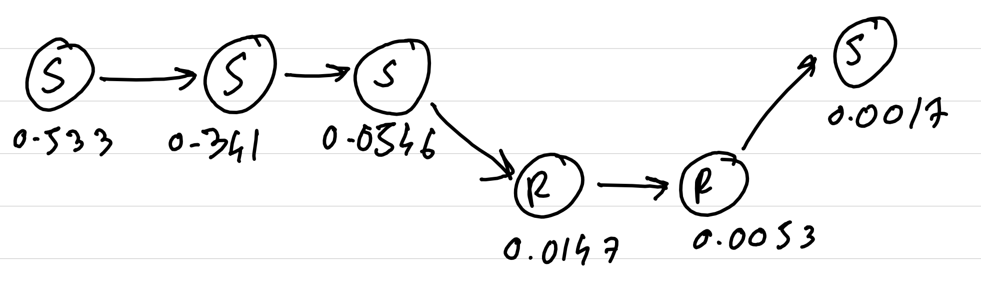
0.333 × 0.2 × 0.4
= 0.043 ✓

0.133 × 0.6 × 0.4
= 0.032

0.341 ✓
0.043

∴ Sunney day

Finally we get



① $S = \{ Hot, Cold \}$ Day type — Hidden states

$V = \{ V_1, V_2, V_3 \}$ No of icecreames consumed — Output states

Example sequence
$x_1 = V_2$
$x_2 = V_3$
$x_3 = V_1$
$x_4 = V_2$
} 4 days data

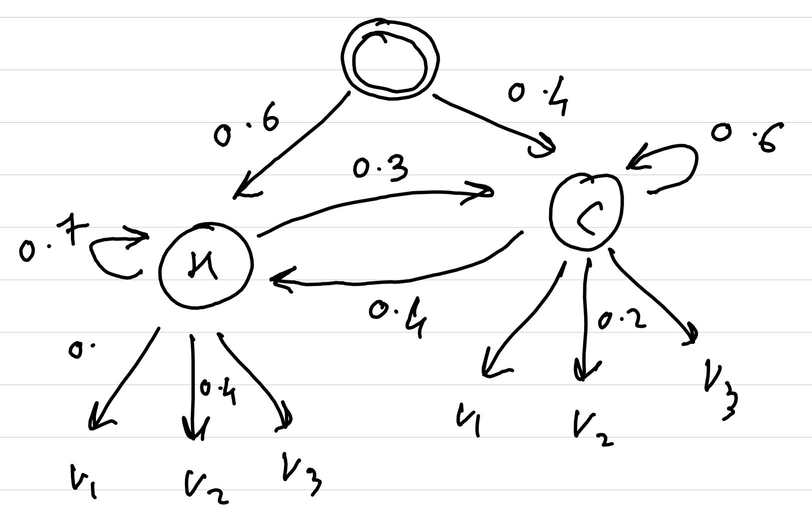
$$A = \begin{array}{c|cc} & H & C \\ \hline H & 0.7 & 0.3 \\ C & 0.4 & 0.6 \end{array}$$ Transmission Matrix

$$B = \begin{array}{c|ccc} & V_1 & V_2 & V_3 \\ \hline H & 0.1 & 0.4 & 0.5 \\ C & 0.7 & 0.2 & 0.1 \end{array}$$ emission Matrix

$$\pi = \begin{array}{cc} H & C \\ 0.6 & 0.4 \end{array}$$ initial state

Find probability that sequence x will be recorded

→



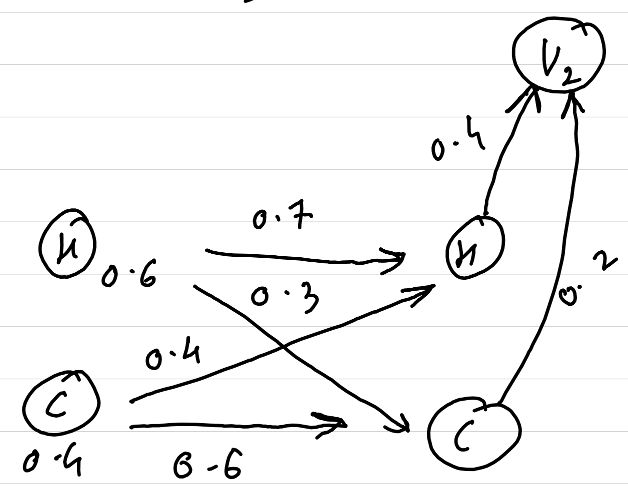
Day 0

(H) 0.6

(C) 0.4

} Prior probabilities π

Day 1



For first sequence $x_1 = V_2$

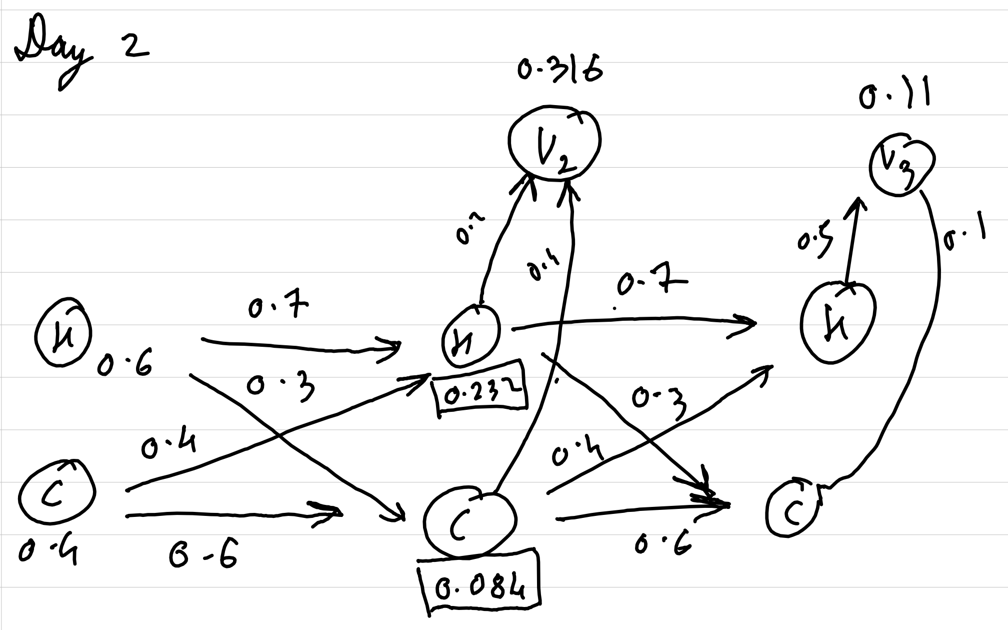
$H \rightarrow (0.6 \times 0.7 \times 0.4) + (0.4 \times 0.4 \times 0.4) = 0.232$

$C \rightarrow (0.4 \times 0.6 \times 0.2) + (0.6 \times 0.3 \times 0.2) = \underline{0.084}$

0.316

$\therefore 0.316$ is probability of buying 2 icecremes on day 1

Day 2



$x_2 = V_3$

$\therefore$ $H = 0.232 \times 0.7 \times \cdots + 0.232 \times \cdots$

$C = 0.084 \times 0.6 \times \cdots \quad + 0.084 \times \cdots$

$\Sigma = \overline{\overline{0.11}}$

and so on

We get
$V_1 = 0.316$
$V_2 = 0.11$
$V_3 = 0.03296$
$V_4 = 0.00966$

This is the probability of the sequences day wise

finally Probability of sequence is

$P(x_1) \times P(x_2) \times \cdots\cdots$

$= 0.316 \times 0.11 \times \cdots\cdots$

This is the probability that sequence $V_1 V_2 V_3 V_4$ will be recorded

"Evaluation problem"

Note

We are **NOT** finding a random walk (unlike Markov chains)

In random walk, probabilities are calculated only from the past inputs

In evaluation problem, we are given the X dataset (eg no of items eaten)

Probability is of day | X. Hence the probability of outputs is also included

In decoding problem, X is given also the path is to be found out

Hence the most likely paths are kept the paths are not added.

Evaluation → chosen

Decoding → Added

HMM Learning (Training)

HMM can be used in Supervised as well as unsupervised scenerios

Supervised → Data consists of sequences of observations along with the corresponding sequences of hidden states

Temperature, icecreames → known

Unsupervised → No info about hidden states only observed sequences

Only icecreames known

There are many methods for HMM learning

① Maximum likelyhood estimation → (Unsupervised)

estimate parameters that maximize the likelyhood of the observed data sequence

Baum Welch algorithms are used

② Expectation Maximization algorithm based training (Unsupervised)

Initialize A, B, π

for an observed sequence O

E: Calculate probabitly at being in state S at time t given O

M: Update transition probabilities

③ Viterbi training - when true state sequence is known or can be estimated (Supervised)

④ Gradient based optimizations

Adjust parameters based on gradient descent on target functions

(Supervised, Unsupervised)

Advantages → 1. Flexibility
2. Efficiency (low cost)
3. Interpretable

Disadvantages → 1. Markov property may not hold always
2. Overfitting
3. Not as robust as NN

Applications → 1. Gene Prediction
2. NLP (POS tagging)
3. SLAM (Robotics)
4. Speech recognition