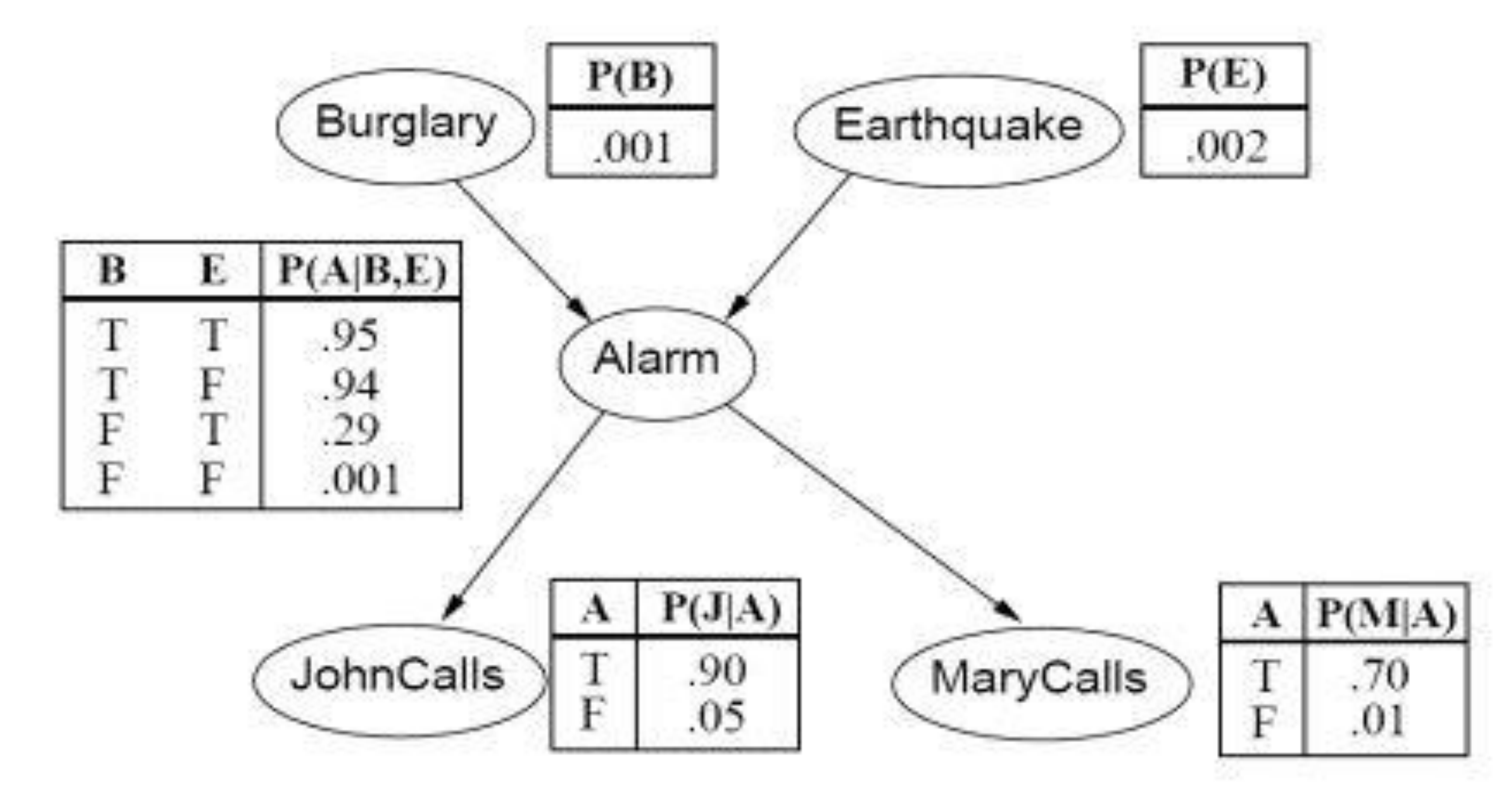
DYNAMIC BAYESIAN NETWORK

AI-2002

OVERVIEW OF BAYESIAN NETWORK

- **Bayesian networks** are a type of probabilistic graphical model that uses Bayesian inference for probability computations. Bayesian networks aim to model conditional dependence, and therefore causation, by representing conditional dependence as edges in a directed graph. Through these relationships, one can efficiently conduct inference on the random variables in the graph using factors.
- Using the relationships specified by the Bayesian network, we can obtain a compact, factorized representation of the joint probability distribution by taking advantage of conditional independence.
- Hence, we say that a Bayesian network is a **directed acyclic graph** in which each edge corresponds to a conditional dependency, and each node corresponds to a unique random variable. Formally, if an edge (A, B) exists in the graph connecting random variables A and B, it means that $P(B|A)$ is a **factor** in the joint probability distribution, so we must know $P(B|A)$ for all values of B and A in order to conduct inference.



BASIC EXAMPLE OF A BAYESIAN NETWORK

WHAT DOES DYNAMIC BAYESIAN NETWORKS DO

ABOUT DYNAMIC BAYESIAN NETWORKS ...

- We have seen that Bayesian Networks are directed acyclic graphs whose edges indicate the conditional dependencies between nodes.
- Dynamic Bayesian networks (DBNs) are an extension of Bayesian networks to model dynamic processes
- A DBN consists of a series of time slices that represent the state of all the variables at a certain time, t.
- For each temporal slice, a dependency structure between the variables at that time is defined, called the base network
- Additionally, there are edges between variables from different slices, with their directions following the direction of time, defining the transition network

TYPES OF BAYESIAN NETWORK MODELS FOR DYNAMIC PROCESSES

- **State Based Models** : State Based Models represent the state of each variable at discrete time intervals, Indicating different time slices, where each time slice indicates the value of each variable at time t.

- **Event/Temporal based Models**: While Even Based Models represent the changes in state of each state variable, hence each temporal node correspond to the time in which a state change occurs.

HOW DOES A DYNAMIC BAYESIAN NETWORK WORK

They contain multi variable nodes and it is useful as it makes use of temporal nodes as a simplified structure, and makes inference faster , also easier implementation.They model events that include both, temporal and ambient aspects.

It consists of a series of time slices that represent the state of all the variables at a certain time t, and for each temporal slice a dependency structure between the variables at that time is the base network.

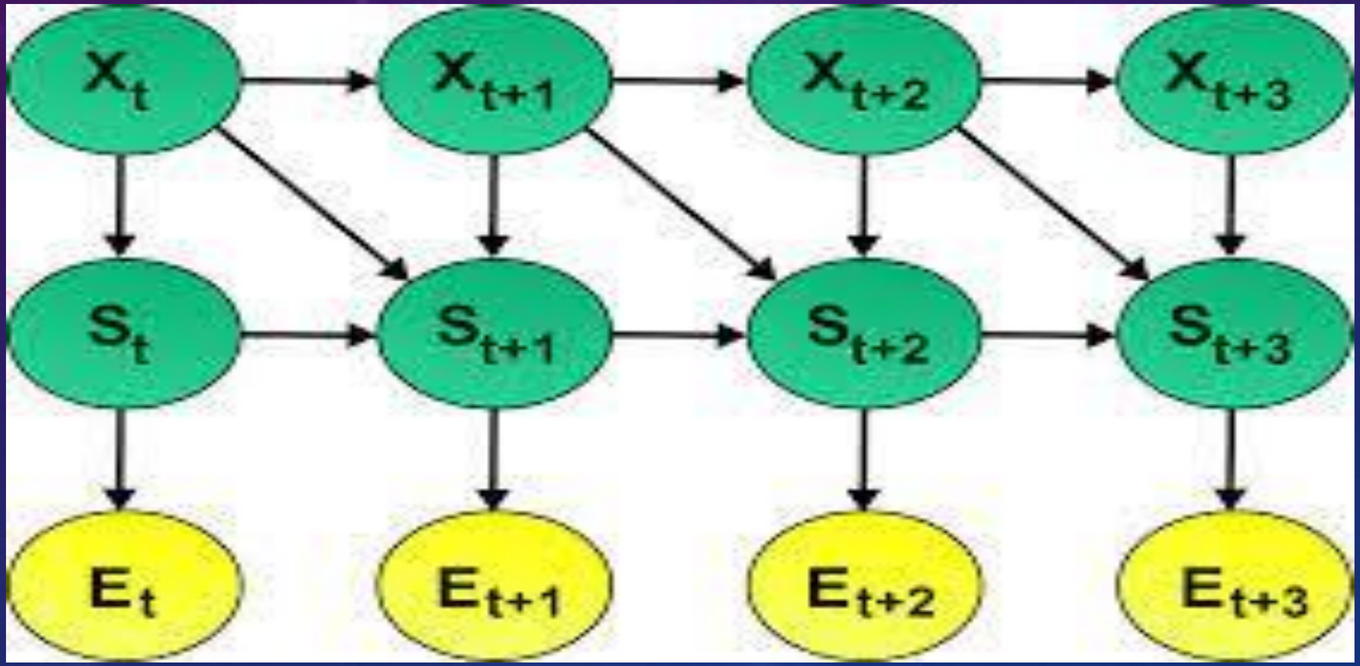
Hence, There are edges between varaibles from different slices, with their direcitons following the direction of time, which defines the transition network.

The state variables at time t depend only on the state variables at time t – 1 and the other variables at time t and the structure and parameters of the model do not change over time.

While making inference, we consider different cases as follows -> Filtering is where we predict the next state based on past observations, Prediction is that type of inference where we predict the future states based on the past observations

Two other types of inferences that we consider would be of Smoothing where we estimate the current state based on past and future observations which comes in very handy for models based on learning, and the other type of inference is Decoding where we find the most likely sequence of hidden variables given the observations,

DBN EXAMPLE



DYNAMIC BAYESIAN NETWORK

- Nodes
  - Temporal Nodes
  - Non – Temporal Nodes
- Example Networks
  - Hidden Markov Model
  - Kalman Filter
- Related Domains
  - Machine Translation
  - Gene Production

HOW DO THE DIFFERENT NODES WORK

Temporal Nodes

- Temporal nodes tend to change their values at different time steps.
- These nodes are used to model both dynamic and static states in different models.
- The variables present in these nodes tend to change values depending on the previous time steps states.
- These tend to be used in Temporal Bayesian Network or Dynamic Bayesian Networks.

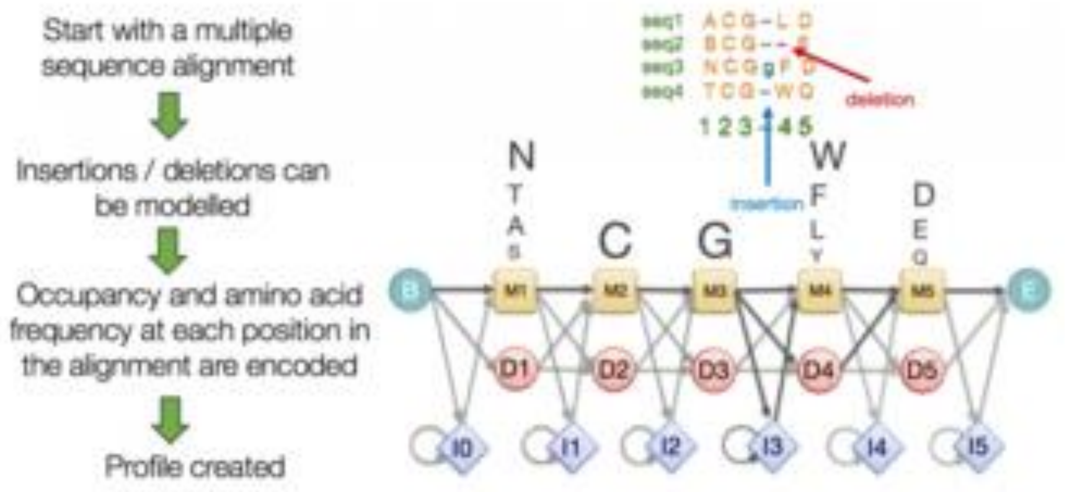
Non – Temporal Nodes

- Non – temporal nodes values stay the same for different time steps.
- These nodes can only model static states in a particular model.
- The variables present in these nodes don't tend to change their value with respect to time but change depending on some action taken place between different states.
- These are not that much used in Temporal Bayesian Network or Dynamic Bayesian Network.

CURRENT STATE OF THE ART

- Common in Robotics and have been used in Speech Recognition, digital Forensics, Protein sequencing and bioinformatics, as DBN's are used to model many uncertain and probabilistic conditions and exploited in different expert systems to model complex interaction among causes and consequences.
- Dynamic Bayesian Networks are used for dynamic gesture recognition and temporal event networks are used for predicting HIV mutational pathways. DBNs provide an alternative to HMMs for dynamic gesture recognition and here we use a particular type of DBN known as dynamic Bayesian Network Classifier (DBNC).
- Many implementations of DBN's include Hidden Markov Model(widely used for tasks describing time varying patterns), Fuzzy Logic, Kalman Filter, etc.

BRIEF OVERVIEW OF HIDDEN MARKOV MODEL



HIDDEN MARKOV MODELS

- Hidden Markov Model are Statistical Model in which the system being modeled is assumed to be in Markov Process.
- Considering these nodes, the goal is to learn about a particular node X by observing another node Y. HMM stipulates that, for each time instance N_0, the conditional probability distribution of Y_{N_0}.
- The Forward Backward Algorithm used in HMM was first described in 1960. One of the first application of HMMs began in speech recognition and later HMMs began to be applied in the analysis of biological sequences in particular DNA, and hence have been ubiquitously used in the field of bioinformatics.
- The parameter learning task is to find, given an output sequence the best set of state transition and emission probabilities, i.e. to derive maximum likelihood estimate of the parameters of the HMM.
- They are known for their applications in thermodynamics, statistical mechanics, and pattern recognition such as Speech, Handwriting , Gesture Recognition and Bioinformatics

GENERAL STRUCTURE OF A HIDDEN MARKOV MODELS

- DBNs can be seen as a generalization of Markov chains:

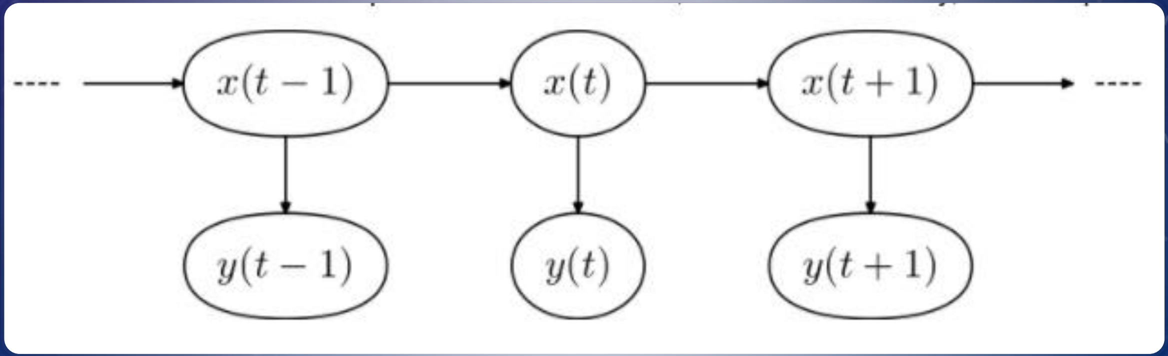
$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2 \mid X_1)\ldots P(X_T \mid X_{T-1}) \quad (1)$$

- And HMMs:

$$P(\{S_{1:T}, Y_{1:T}\}) =$$

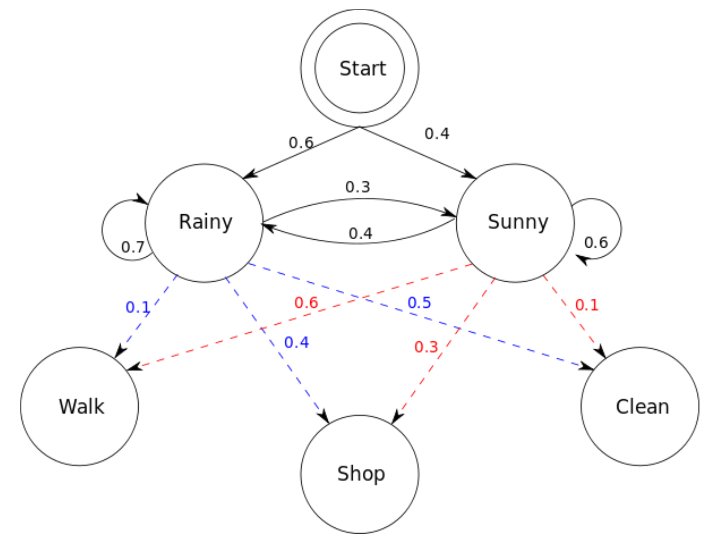
$$P(S_1)P(Y_1 \mid S_1)\ {}^{T}\Pi_{t=2}\ P(S_t \mid S_{t-1})P(Y_t \mid S_t) \quad (2)$$

- Kalman Filters, also have one state and one observation variable, but both variables are continuous. The basic Kalman filter assumes Gaussian distributions and linear functions for the transitions and observations



WEATHER GUESSING GAME HMM EXAMPLE MODEL

- Start Probability -> Agent X's probability of knowing a particular states weather at that timestep
- Transition Probability -> represents the probability of change in weather, in a state underlying Markov Chain.
- Emission Probability -> represents Agent X 's Probability of doing an action in the next state depending on previous state's weather.




```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
   }

emission_probability = {
   'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
   'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
   }
```

HMM

HMM

→ It is statistical model which generate
~~based on~~ Models generatio based on some Input Sequence.
If you have Inputs it will gives you New states on which we will perform transitions' (S1, S2, S3 →)

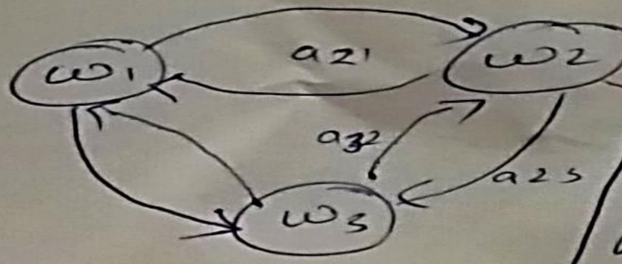
Properties of HMM mel
* → Much required ⇒ Markov Property
↓
Memory less.
Considering only 1 state.

→ Sole part (Future prediction depend only on present state.
(neither paste nor Future)

$\boxed{\text{HMM}}$ → Finite State Machine
→ some no: of states.
① Hidden State (ω)
② Visible / output state (V)

→ hidden strat

a_{12} → Transition Probability



b_{21} v_1 living from
b_{22} v_2 state to the
b_{23} v_3 other

→ output o_i (visible state).

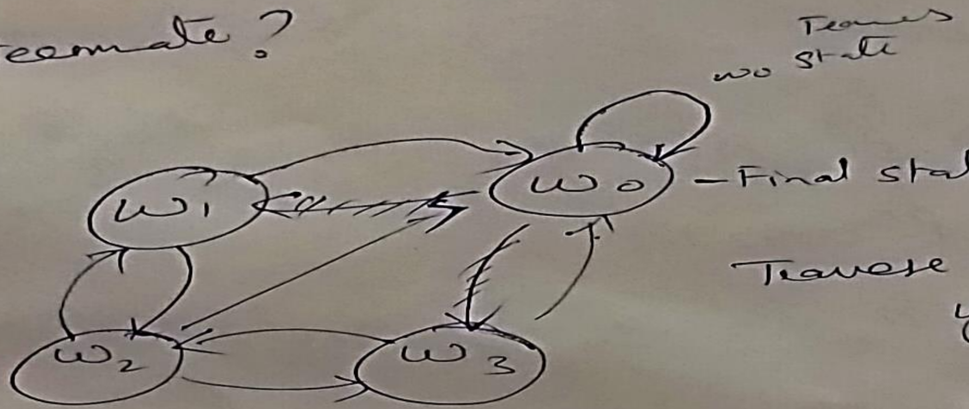
Probability 0 - 1

emission probability ← weight associated with ~~wht~~ states

O/P from particular states have weight.

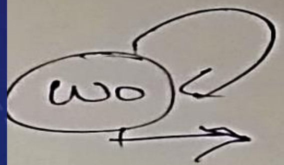
$$\sum_j a_{ij} = 1 \qquad \sum_k b_{jk} = 1$$

When to terminate?



— Final state

Traverse (One day you will reach ω_0).



Transition
$a_{00} = 1$

$b_{00} = 1$

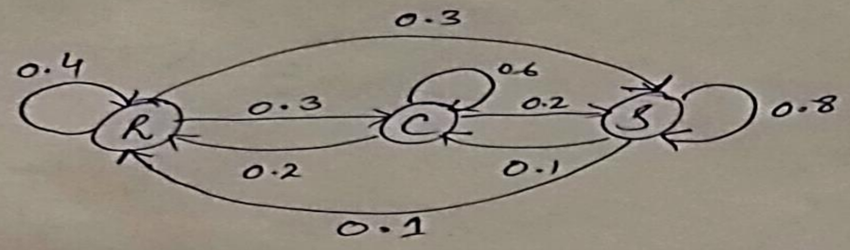
V_0 output

EXAMPLE

Example:- To predict weather pattern for next 7 days sequence as

{Sun, Sun, Rain, Rain, Sun, Cloudy, Sun}

- Today is Sunny.



Transition Matrix

	R	C	S
R	0.4	0.3	0.3
C	0.2	0.6	0.2
S	0.1	0.1	0.8

$S_1 = R, \; S_2 = C, \; S_3 = S$

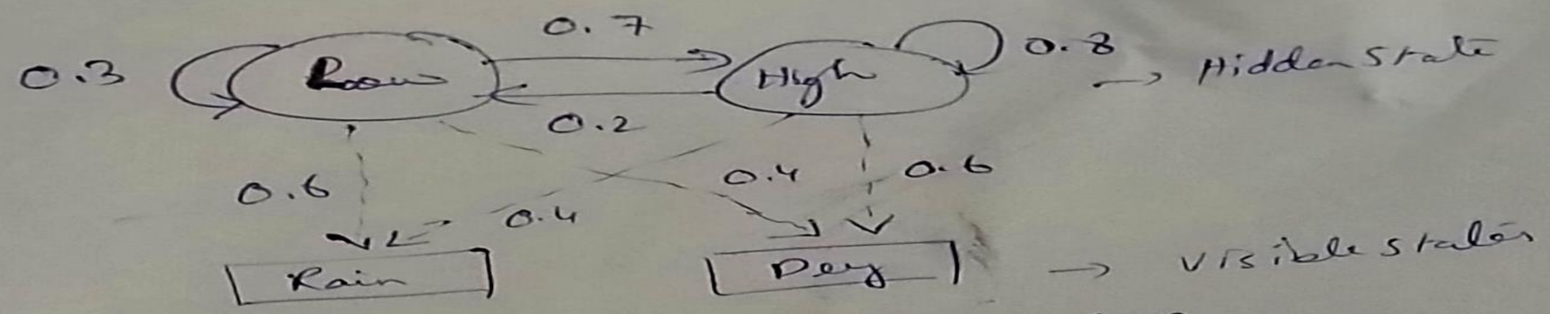
$P(S_3) \cdot P(S_3/S_3) \cdot P(S_3/S_3) \cdot P(S_1/S_3) \cdot P(S_1/S_1) \cdot P(S_3/S_1) \cdot$
$P(S_2/S_3) \cdot P(S_3/S_2)$

$\pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \cdot \cancel{a_{2} \cdot a}$

$0.1\,(0.8)\,(0.8)\,(0.1)\,(0.4)\,(0.3)\,(0.1)\,(0.2)$

$= 1.536 \times 10^{-4}$

HMM Example:- 2



2- Initial Probabilities $\pi_2 = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}$ $\begin{matrix} \text{Low} \\ \text{High} \end{matrix}$

Sequence to Calculate :- [Dry, Rain]

Sol:- Consider all possible hidden states combinations

P(Dry, Rain) = P[(Dry, Rain), (low, low)] + P[(Dry, Rain), (low, High)] +
P[(Dry, Rain), (High, low)] + P(Dry, Rain), (High, High)).

1st term = P[(Dry, Rain), (low, low)]
= P(Dry/low) · P(Rain/low) · P(low) · P(low/low).
0.4 × 0.6 × 0.4 × 0.3
= 0.288

2nd term = 0.0448
3rd term = 0.043
4th term = 0.115

P(Dry Rain) = 0.288 + 0.0448 + 0.043 + 0.115
= 0.491