REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider-BISKRA.

Faculté des sciences exactes et des sciences de la nature et de la vie

Département d'informatique

Machine Learning (ML)

Dr. TIBERMACINE Ahmed

CONTENT SUMMARY

Chapter 5:
Hidden Markov Models "HMMs"

INTRODUCTION

- The chapter explores Hidden Markov Models (HMMs) by building on stochastic processes and Markov chains.
- Stochastic processes establish the basis for randomness, Markov chains focus on state transitions, and HMMs introduce hidden states influencing observable outcomes.
- The chapter covers HMM structure, key algorithms, offering a fundamental understanding of their applications in modeling dynamic systems..

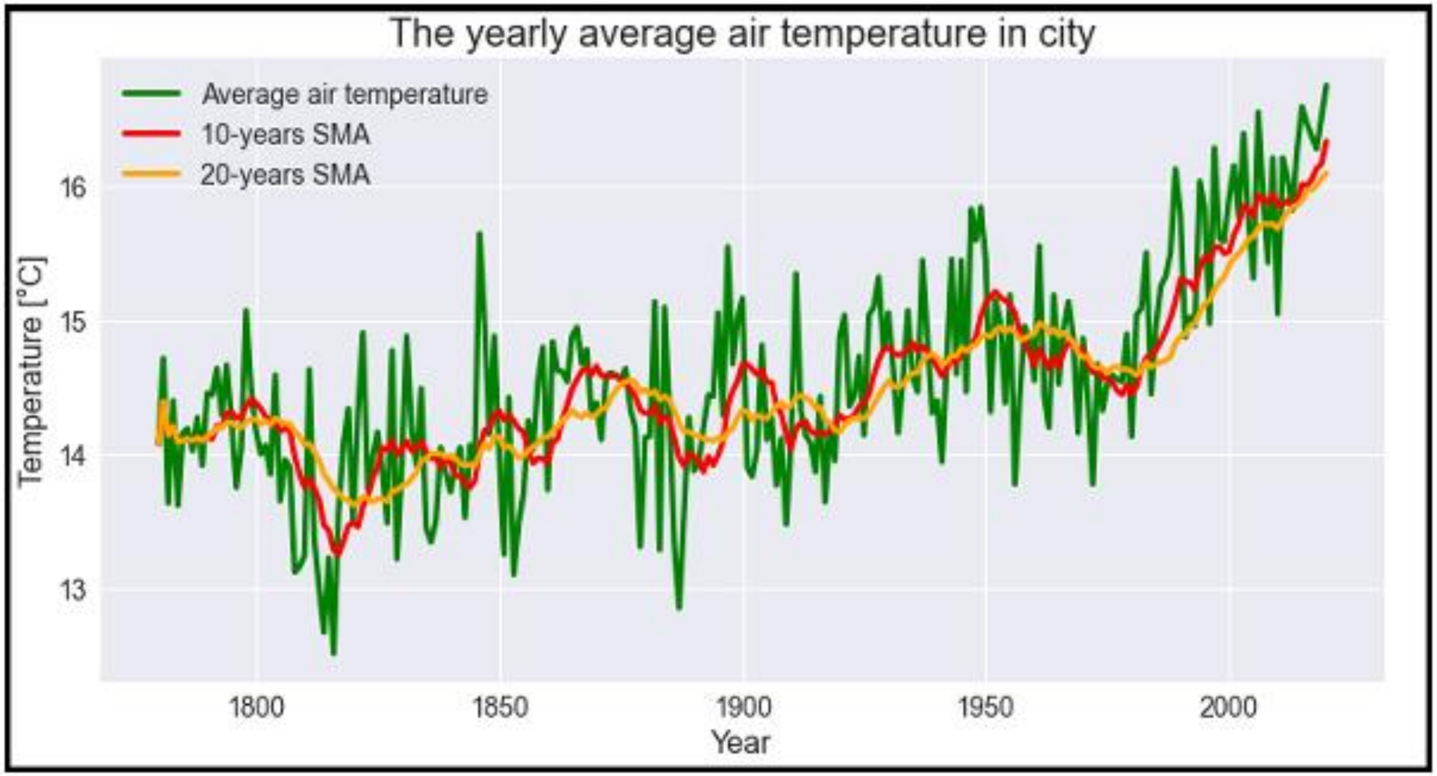
Stochastic Process

❖ Definition

- A stochastic process is a mathematical model that describes the evolution of random variables over time.
- In a stochastic process, the outcome at each time step is not deterministic but is governed by probability.
- It's defined as $\{X(t), t \in T\}$, where $X(t)$ represents the state of the system at time t, and T is the index set, often representing discrete or continuous time.

Stochastic Process

❖ **Example of a Stochastic Process**



Stochastic Process

- **Classification of Stochastic Processes**
  - Discrete-Time vs. Continuous-Time
  - Stationary vs. Non-Stationary
  - Markov vs. Non-Markov
  - Ergodic vs. Non-Ergodic
  - Homogeneous vs. Non-Homogeneous:
- .

Markov model

❖ Overview

- Markov models represent random systems where future states are independent of the past.
- They illustrate all possible states, transitions, rates, and probabilities.
- These models are commonly used to depict state probabilities and transition rates
- The method is generally used to:
  1. model systems.
  2. identify patterns,
  3. make predictions,
  4. and learn the statistics of sequential data.

Markov model

- **Types of Markov models**

1. Markov chain:
2. Hidden Markov model:
3. Markov decision processes:
4. Partially observable Markov decision processes:

Markov model

- **Representation**
- Markov models can be shown using math equations or pictures.

- Graphic Markov models typically use:
  1. circles (each containing states);
  2. directional arrows to show potential transitional changes between them.

- The directional arrows are labelled with the rate or the variable one for the rate

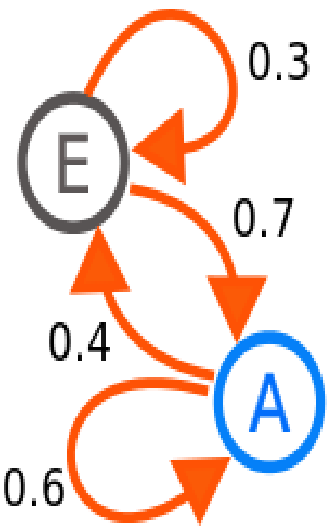
Markov model

- **Applications:**
- modelling languages,
- natural language processing (NLP),
- image processing,
- bioinformatics,
- speech recognition,
- and modelling computer hardware and software systems

Markov chain

❖ **Overview**

- A Markov chain is a math model for systems with events that move from one state to another.
- The next state only depends on the current one, not the ones before.
- A Markov chain, a memoryless stochastic process, predicts the future based only on the current state.
- It's valuable for modeling real-world scenarios like weather, finance, and biology.



MARKOV CHAIN

❖ Key Concepts

- **States & State Space:** a Markov chain consists of a finite or countable set of states X0, X1, X2,… .
- The **state** of a Markov chain is the value of **Xt** at time **t**.
- For example, if **Xt = 6**, we say the process is in state 6 at time t.
- The state space of a Markov chain, often denoted as S = $\{S_1, S_2, S_3, \ldots\}$, is the set of values that each **Xt** can take. For example, S = {1, 2, 3, 4, 5, 6, 7}.

Markov chain

❖ **Key Concepts**

- A **trajectory** of a Markov chain is a particular set of values for X0, X1, X2, ...
- The **transition probability** is the probability of moving from one state of a system into another state.

pij = P(next state Sj at t=1 | initial state Si at t=0)

- **Transition Probability Matrix**:

Next state

$$P = \begin{array}{c} \\ S1 \\ S2 \\ S3 \end{array} \begin{array}{c} \begin{array}{ccc} S1 & S2 & S3 \end{array} \\ \begin{pmatrix} P11 & P12 & P13 \\ P21 & P22 & P23 \\ P31 & P32 & P33 \end{pmatrix} \end{array}$$

Initial state

MARKOV CHAIN

Key Concepts

- **Markov Property:** only the most recent point in the trajectory affects what happens next.

Explanation:

$$\mathbb{P}(X_{t+1} = s \mid X_t = s_t, \underbrace{\cancel{X_{t-1} = s_{t-1}, X_{t-2} = s_{t-2}, \ldots, X_1 = s_1, X_0 = s_0}})$$

distribution of X_{t+1}

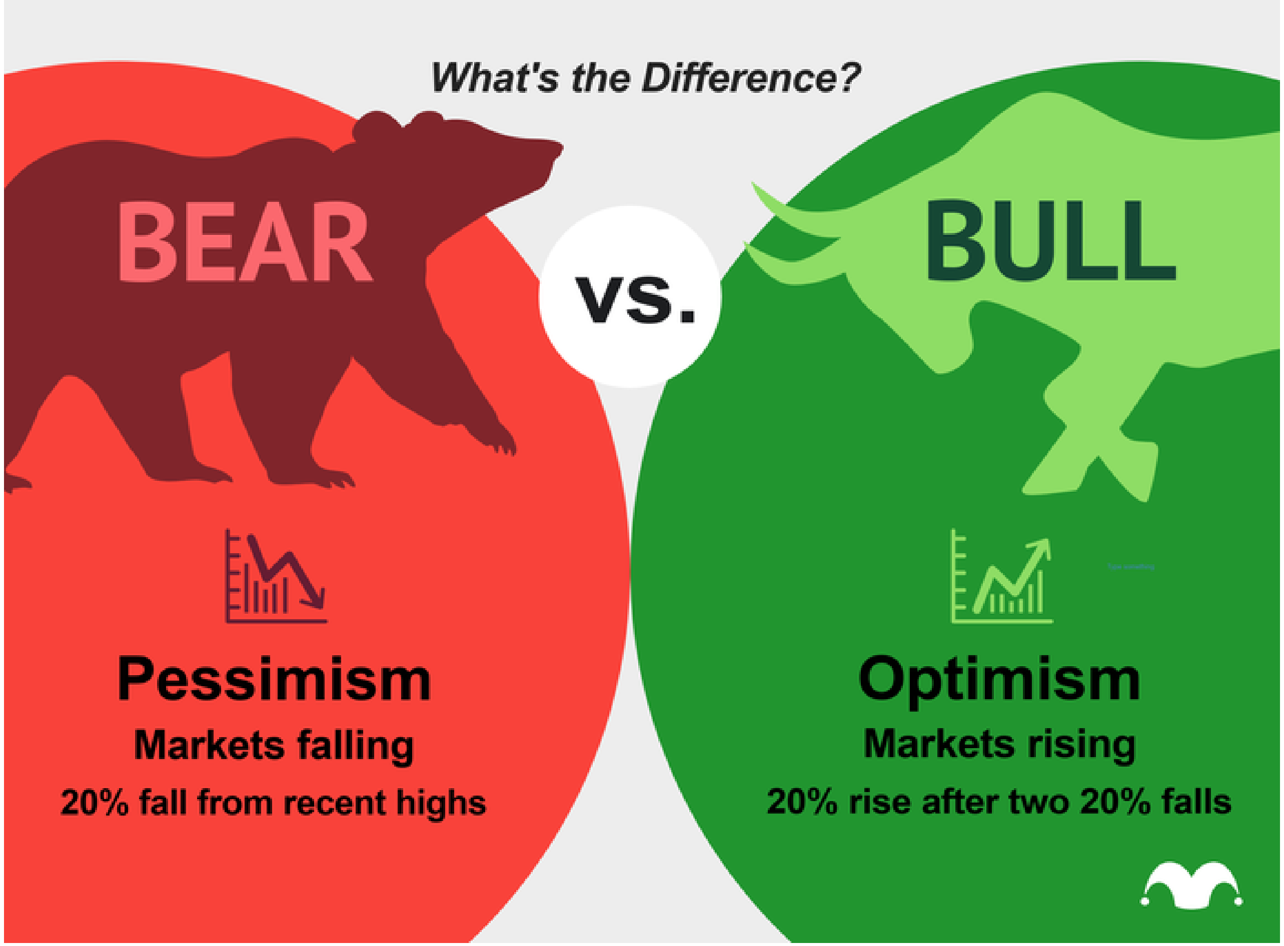
depends on X_t

but whatever happened before time t doesn't matter.

- **Time Homogeneity:** This means the transition probabilities remain constant over time.

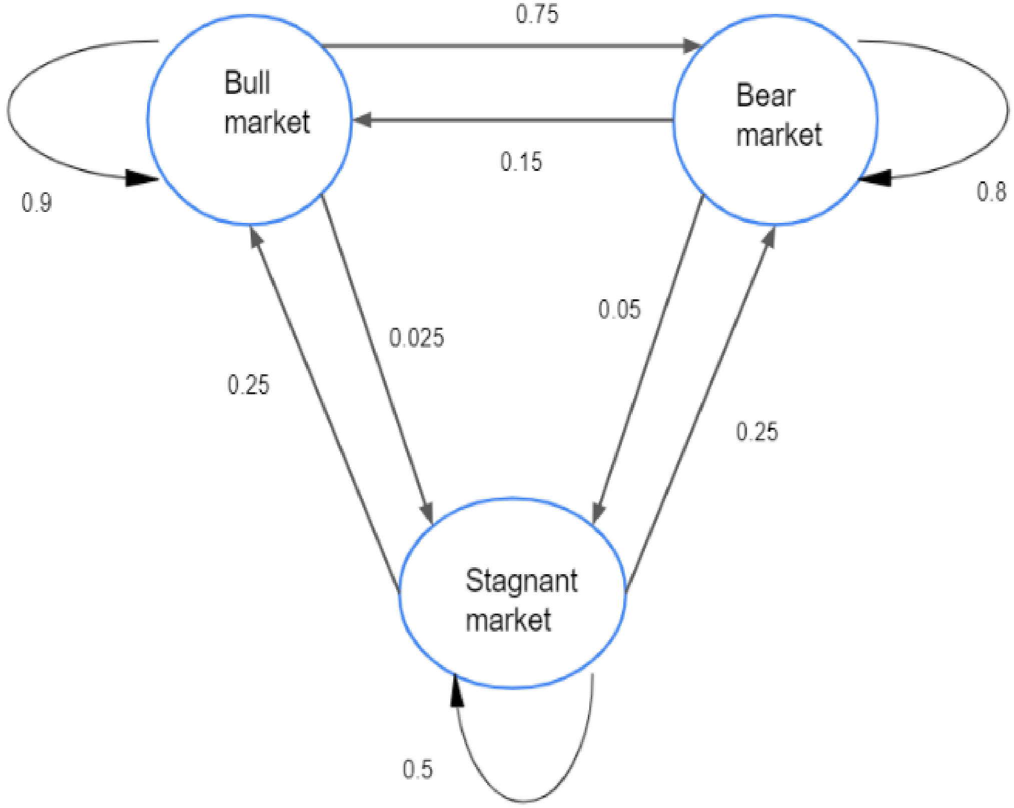
Markov chain

- <u>**Illustrative Example:**</u>
- **Markov chains to predict stock market trends**



Markov chain

- <u>Illustrative Example:</u>
- Markov chains to predict stock market trends



MARKOV CHAIN

- **Illustrative Example:**
- **Markov chains to predict stock market trends**
- By compiling these probabilities into a table, we get the following state transition matrix **M**:

From \ *To*	Bull	Bear	Stagnant
Bull	0.9	0.075	0.025
Bear	0.15	0.8	0.05
Stagnant	0.25	0.25	0.5

$$= \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix} = M$$

- We then create a 1x3 vector C.
- In this example we will choose to set the current week as bearish, resulting in the vector C = [0 1 0].

Markov chain

- **Illustrative Example:**
- **Markov chains to predict stock market trends**

One week from now: $C * M^1 = [0 \quad 1 \quad 0]\begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}^1 = [0.15 \quad 0.8 \quad 0.05]$

5 weeks from now: $C * M^5 = [0 \quad 1 \quad 0]\begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}^5 = [0.48 \quad 0.45 \quad 0.07]$

52 weeks from now: $C * M^{52} = [0 \quad 1 \quad 0]\begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}^{52} = [0.63 \quad 0.31 \quad 0.05]$

99 weeks from now: $C * M^{99} = [0 \quad 1 \quad 0]\begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}^{99} = [0.63 \quad 0.31 \quad 0.05]$

Markov chain

- **Illustrative Example:**
- In a regular Markov chain, after a sufficient number of iterations, all transition probabilities converge to a constant value (**steady state)**
- As iterations increase, the Markov chain converges to steady-state probabilities: 63% bullish, 31% bearish, and 5% stagnant.
- These probabilities are independent of the initial state and can be used for practical purposes.

Hidden Markov models (HMMs)

- **Definition**

- HMMs are a dual stochastic system consisting of an unobservable stochastic process, and an observable process influenced by these hidden states.

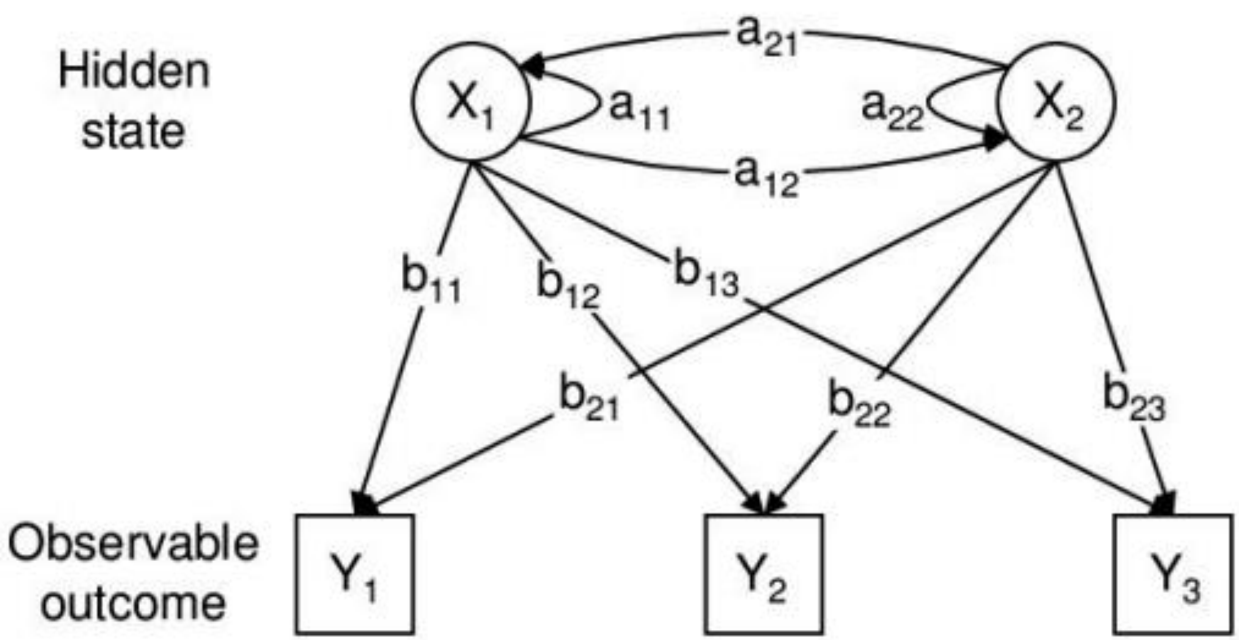
- HMMs aim to reveal hidden information within observables.

- HMMs comprise two key components:
  - Hidden states (X)
  - Observable symbols (Y).

HIDDEN MARKOV MODELS (HMMS)

❖ **Hmms Components**

- These components are governed by essential probabilities:
  - Transition probabilities (a):
  - Emission probabilities (b):
  - Initial probability distribution (X_i):



Hidden Markov models (HMMs)

- **Applications of HMCs**
  - Speech Recognition:
  - Natural Language Processing:
  - Finance:
  - Bioinformatics:

Hidden Markov models (HMMs)

❖ **Fundamental HMM Algorithms**

❖ **Computing Sequence Probabilities: The Forward Algorithm**

- The Forward Algorithm in HMMs calculates the probability of observing a sequence given the model.
- It is commonly used for problems like sequence likelihood estimation

Hidden Markov models (HMMs)

❖ **Fundamental HMM Algorithms**

- **Computing Sequence Probabilities: The Forward Algorithm**
- **Inputs:**

1. **Hidden Markov Model (HMM) parameters:**
   - Hidden States (S):.
   - Observations (O):
   - Initial State Probabilities (π):.
   - Transition Probabilities (A):
   - Emission Probabilities (B):.
2. **Sequence of observations (O_1, O_2, ..., O_x):**

HIDDEN MARKOV MODELS (HMMS)

❖ **Fundamental HMM Algorithms**

- **Computing Sequence Probabilities: The Forward Algorithm**
- **Steps:**

1. **Initialization:**

- Initialize an array of forward probabilities (α) for each hidden state and the first observation.
- For each hidden state S_i:

$$\alpha(1, S_i) = \pi(S_i) * B(S_i, O_1)$$

Where: $\pi(S_i)$ is the initial state probability and

$B(S_i, O_1)$ is the emission probability for the first observation.

Hidden Markov models (HMMs)

- **Fundamental HMM Algorithms**
- **Computing Sequence Probabilities: The Forward Algorithm**
- **Steps:**

2. Recursion:

- For each time step t from 2 to T (T is the length of the observation sequence):

  Calculate the forward probabilities for each hidden state S_i at time t:

$$\alpha(t, S_i) = \sum_{all\ Sx} [\alpha(t-1, S_x) * A(S_x, S_i) * B(S_i, O_t)]$$

- This calculation involves considering the probability of being in state S_x at time t-1 (α(t-1, S_x)), transitioning from S_x to S_i (A(S_x, S_i)), and emitting observation O_t (B(S_i, O_t)).
- The sum is taken over all possible states S_x, representing the contribution of each state to the probability of being in state Si at time t.

Hidden Markov models (HMMs)

- **Fundamental HMM Algorithms**
- **Computing Sequence Probabilities: The Forward Algorithm**
- **Steps:**

3. Termination:

- The final probability is obtained by summing the forward probabilities for all hidden states at the last time step T:

$$P(\text{Observations}) = \sum \alpha(T, S_i), \text{ for all } S_i.$$

- P(Observations): represents the probability of observing the given sequence of observations under the HMM with its specific parameters.

HIDDEN MARKOV MODELS (HMMS)

❖ **Fundamental HMM Algorithms**

- **Finding the Most Likely State Sequence: The Viterbi Algorithm**
- The Viterbi Algorithm is used in HMMs to find the most likely sequence of hidden states given observations.
- **Inputs:**
  - Hidden States (S):
  - Observations (O):.
  - Initial State Probabilities (π):
  - Transition Probabilities (A):
  - Emission Probabilities (B):

Hidden Markov models (HMMs)

- **Fundamental HMM Algorithms**
- **Finding the Most Likely State Sequence: The Viterbi Algorithm**
- **Steps:**

1. Initialization:

- Initialize the Viterbi probabilities (δ) for the first observation (O[1]) and the backpointer (ψ) for each state:

$$\delta(1, i) = \pi(i) * B(i, O[1]) \text{ for all states } i.$$

- ψ(1, i) is undefined at this point.

Hidden Markov models (HMMs)

❖ Fundamental HMM Algorithms

- **Finding the Most Likely State Sequence: The Viterbi Algorithm**
- **Steps:**

2. Recursion:

- Calculate the Viterbi probabilities and backpointers for the remaining observations (O[2] to O[T]) for each state and each time step:
- For each time step t from 2 to T:
- For each state j:
- Calculate $\delta(t, j)$ and $\psi(t, j)$ as follows:

$$\delta(t, j) = \max[\delta(t-1, i) * A(i, j) * B(j, O[t])], \text{ where i is a state in S.}$$

$$\psi(t, j) = \operatorname{argmax}[\delta(t-1, i) * A(i, j)], \text{ where i is a state in S.}$$

Hidden Markov models (HMMs)

- **Fundamental HMM Algorithms**
- **Finding the Most Likely State Sequence: The Viterbi Algorithm**
- **Steps:**

3. Termination:

- Calculate the overall probability of the most likely path and the final state in that path:

$$P^*(O) = \max[\delta(T, i)], \text{ where } i \text{ is a state in } S.$$

- The most likely final state is the state j that maximizes $\delta(T, j)$.

4. Backtracking:

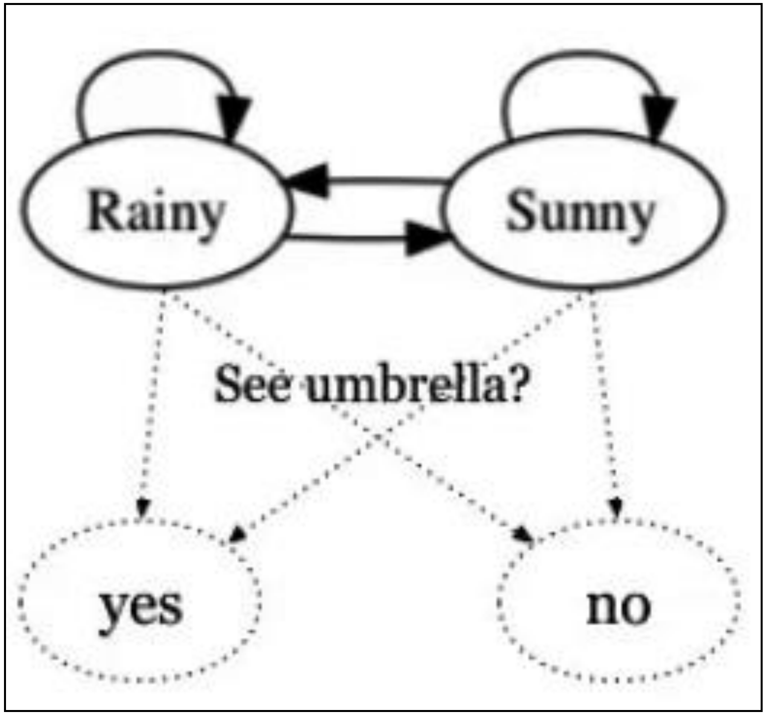
- Backtrack through backpointers, starting from the final state found in Step 3, to recover the most likely sequence of hidden states.
- The resulting sequence best explains the given observations in Hidden Markov Models.

HIDDEN MARKOV MODELS (HMMS)

❖ **Illustrative example:**

❖ **Weather Prediction with Hidden Markov Models**

- **HMM Setup:**
  - Hidden States: Sunny (S), Rainy (R)
  - Observations: Umbrella (U), No Umbrella (N)



Hidden Markov models (HMMs)

❖ **Illustrative example: Weather Prediction with Hidden Markov Models**

- **Model Parameters:**
- Initial State Probabilities (π):
- P(S) = 0.7
- P(R) = 0.3
- Transition Probabilities (A):
- P(S | S) = 0.8
- P(R | S) = 0.2
- P(S | R) = 0.4
- P(R | R) = 0.6
- Emission Probabilities (B):
- P(U | S) = 0.2
- P(N | S) = 0.8
- P(U | R) = 0.6
- P(N | R) = 0.4

HIDDEN MARKOV MODELS (HMMS)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**

- **Questions:**
- What is the probability of observing the sequence U, U, N, U using the Forward Algorithm?
- What is the most likely sequence of hidden states for the observations U, U, N, U using the Viterbi Algorithm

HIDDEN MARKOV MODELS (HMMS)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**

- **Solution: Using the Forward Algorithm:**
- **Step 1: Initialization:**
  - α(1, S) = P(U | S) * P(S) = 0.2 * 0.7 = 0.14
  - α(1, R) = P(U | R) * P(R) = 0.6 * 0.3 = 0.18
- **Step 2: Recursion:** (U, N, U)
  - α(2, S) = [α(1, S) * P(S | S) * P(U | S)] + [α(1, R) * P(S | R) * P(U | S)] = [0.14 * 0.8 * 0.2] + [0.18 * 0.4 * 0.2] = 0.0224 + 0.0144 = 0.0368
  - α(2, R) = [α(1, S) * P(R | S) * P(U | R)] + [α(1, R) * P(R | R) * P(U | R)] = [0.14 * 0.2 * 0.6] + [0.18 * 0.6 * 0.6] = 0.0168 + 0.0648 = 0.0816

Hidden Markov models (HMMs)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**
- **Solution:** **Using the Forward Algorithm:**
  - α(3, S) = [α(2, S) * P(S | S) * P(N | S)] + [α(2, R) * P(S | R) * P(N | S)] = [0.0368 * 0.8 * 0.8] + [0.0816 * 0.4 * 0.8] = 0.023552 + 0.025984 = 0.049536
  - α(3, R) = [α(2, S) * P(R | S) * P(N | R)] + [α(2, R) * P(R | R) * P(N | R)] = [0.0368 * 0.2 * 0.4] + [0.0816 * 0.6 * 0.4] = 0.002944 + 0.019584 = 0.022528
  - α(4, S) = [α(3, S) * P(S | S) * P(U | S)] + [α(3, R) * P(S | R) * P(U | S)] = [0.049536 * 0.8 * 0.2] + [0.022528 * 0.4 * 0.2] = 0.007936 + 0.00180576 = 0.00974176
  - α(4, R) = [α(3, S) * P(R | S) * P(U | R)] + [α(3, R) * P(R | R) * P(U | R)] = [0.049536 * 0.2 * 0.6] + [0.022528 * 0.6 * 0.6] = 0.00593632 + 0.00814464 = 0.01408196

Hidden Markov models (HMMs)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**
- **Solution: Using the Forward Algorithm:**
- **Step 3: Termination:**
- Calculate the overall probability of observing the entire sequence U, U, N, U by summing up the forward probabilities at the last time step (T).

$$P(\text{Observations}) = \alpha(4, S) + \alpha(4, R)$$

$$= 0.00974176 + 0.01408196 = 0.02382372$$

- the probability of the sequence U, U, N, U is approximately 0.0238.

HIDDEN MARKOV MODELS (HMMS)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**
- **Solution:. Using the Viterbi Algorithm**
- **Step 1: Initialization:**

  - $\delta(1, S) = P(U \mid S) * P(S) = 0.2 * 0.7 = 0.14$
  - $\delta(1, R) = P(U \mid R) * P(R) = 0.6 * 0.3 = 0.18$
- No need for a backpointer at this step.

Hidden Markov models (HMMs)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**

- **Solution:**. **Using the Viterbi Algorithm**
- **Step 2: Recursion:** Calculate the Viterbi probabilities and backpointers for the remaining observations (U, N, U) for both states.
- For observation U at time step 2:
- δ(2, S) = max[δ(1, S) * P(S | S) * P(U | S), δ(1, R) * P(S | R) * P(U | S)] = max[0.14 * 0.8 * 0.2, 0.18 * 0.4 * 0.2] = max[0.0224, 0.0144] = 0.0224 (maximum from S)
- ψ(2, S) = argmax[δ(1, S) * P(S | S) * P(U | S), δ(1, R) * P(S | R) * P(U | S)] = S
- δ(2, R) = max[δ(1, S) * P(R | S) * P(U | R), δ(1, R) * P(R | R) * P(U | R)] = max[0.14 * 0.2 * 0.6, 0.18 * 0.6 * 0.6] = max[0.0168, 0.0648] = 0.0648 (maximum from R)
- ψ(2, R) = argmax[δ(1, S) * P(R | S) * P(U | R), δ(1, R) * P(R | R) * P(U | R)] = R

Hidden Markov models (HMMs)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**
- **Solution:. Using the Viterbi Algorithm**
- For observation N at time step 3:
- δ(3, S) = max[δ(2, S) * P(S | S) * P(N | S), δ(2, R) * P(S | R) * P(N | S)] = max[0.0224 * 0.8 * 0.8, 0.0648 * 0.4 * 0.8] = max[0.014336, 0.020736] = 0.020736 (maximum from R)
- ψ(3, S) = argmax[δ(2, S) * P(S | S) * P(N | S), δ(2, R) * P(S | R) * P(N | S)] = R
- δ(3, R) = max[δ(2, S) * P(R | S) * P(N | R), δ(2, R) * P(R | R) * P(N | R)] = max[0.0224 * 0.2 * 0.4, 0.0648 * 0.6 * 0.4] = max[0.001792, 0.019584] = 0.019584 (maximum from R)
- ψ(3, R) = argmax[δ(2, S) * P(R | S) * P(N | R), δ(2, R) * P(R | R) * P(N | R)] = R

HIDDEN MARKOV MODELS (HMMS)

- **Illustrative example:**
- **Weather Prediction with Hidden Markov Models**

- **Solution:**. **Using the Viterbi Algorithm**
- For observation U at time step 4:
- δ(4, S) = max[δ(3, S) * P(S | S) * P(U | S), δ(3, R) * P(S | R) * P(U | S)] = max[0.020736 * 0.8 * 0.2, 0.019584 * 0.4 * 0.2] = max[0.00331776, 0.00156672] = 0.00331776 (maximum from S)
- ψ(4, S) = argmax[δ(3, S) * P(S | S) * P(U | S), δ(3, R) * P(S | R) * P(U | S)] = S
- δ(4, R) = max[δ(3, S) * P(R | S) * P(U | R), δ(3, R) * P(R | R) * P(U | R)] = max[0.020736 * 0.2 * 0.6, 0.019584 * 0.6 * 0.6] = max[0.002496, 0.00669312] = 0.00669312 (maximum from R)
- ψ(4, R) = argmax[δ(3, S) * P(R | S) * P(U | R), δ(3, R) * P(R | R) * P(U | R)] = R

Hidden Markov models (HMMs)

❖ **Illustrative example:**

❖ **Weather Prediction with Hidden Markov Models**

- **Solution:**. **Using the Viterbi Algorithm**
- **Step 3: Termination:** Calculate the overall maximum probability for the last time step (T) and determine the most likely final state (j*):
  - $\delta(4, S) = 0.00331776$
  - $\delta(4, R) = 0.00669312$
- The maximum probability at time step 4 is obtained from state R (Rainy):
  - $j^* = \text{argmax}[\delta(4, j)] = R$

HIDDEN MARKOV MODELS (HMMS)

❖ **Illustrative example:**

❖ **Weather Prediction with Hidden Markov Models**

- **Solution:. Using the Viterbi Algorithm**
- **Step 4: Backtracking:**
- the complete most likely state sequence:
  1. Time Step 4: $j^* = R$ (Rainy)
  2. Time Step 3: $\psi(3, R) = R$ (Rainy)
  3. Time Step 2: $\psi(2, R) = R$ (Rainy)
  4. Time Step 1: $\psi(1, R) = R$ (Rainy)

- So, the most likely sequence of hidden states is: R, R, R, R,