# Markov Models

COMP5318 Machine Learning and Data Mining

semester 2, 2024, week 11

**Nguyen H. Tran (slides prepared by Irena Koprinska)**

Reference: : Witten ch.9.8, Tan ch.4.5

# Outline

- Markov model
- Hidden Markov Model (HMM)
- HMM problem 1: Evaluation
- HMM problem 2: Decoding
- HMM problem 3: Learning
- HMM applications

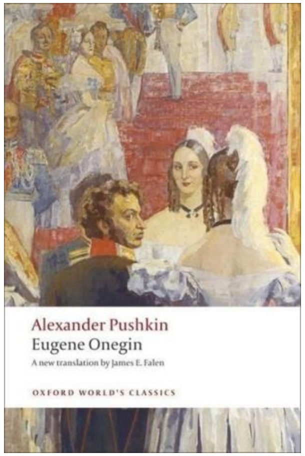
# Markov models

# History of Markov chains

- Markov chains were proposed by the Russian mathematician Andrey Markov
- He used them to analyze the distribution of vowel and consonant letters in Russian poetry - in Pushkin’s poem “Eugene Onegin”
- He meticulously counted the frequencies of pairs of letters
- Transition matrix based on the first 20,000 letters from the poem:

$$P = \begin{array}{r c} & \begin{array}{cc} \text{vowel} & \text{consonant} \end{array} \\ \begin{array}{r} \text{vowel} \\ \text{consonant} \end{array} & \begin{pmatrix} 0.175 & 0.825 \\ 0.526 & 0.474 \end{pmatrix} \end{array}$$

https://en.wikipedia.org/wiki/Andrey_Markov



https://www.amazon.com.au/Eugene-Onegin-Novel-Alexander-Pushkin/dp/0199538646

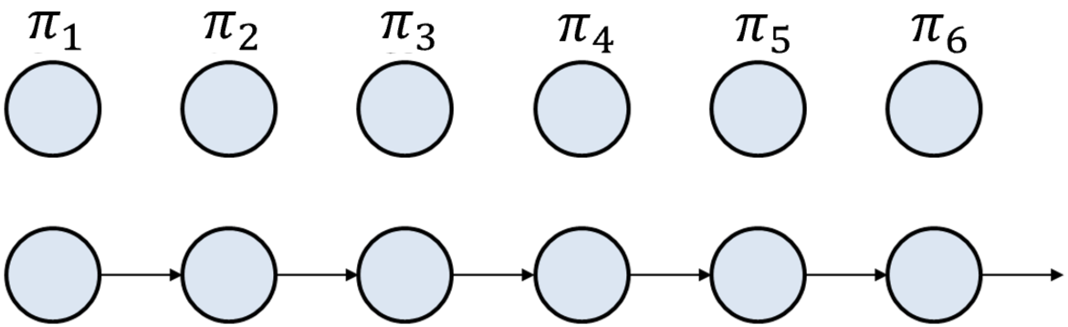
# Markov chain

- A Markov chain (also called Markov process) is a model describing a sequence of transitions from one state to another, in which the probability of each state depends only on the previous state
- More specifically – consider:
  - A finite number of states π, e.g. two states with values *hot* and *cold*
  - Discrete time steps: $t_0, t_1, t_2,\ldots$
  - At each time, $t_i$, the system is in a state $\pi_i$, e.g. $\pi_1 = hot,$ then $\pi_2 = cold,$ etc.
  - Between two subsequent time steps $t_i$ and $t_{i+1}$, the system changes its state from $\pi_t$ to $\pi_{t+1}$. The transition from $\pi_t$ to $\pi_{t+1}$ is probabilistic.
  - The probability of any concrete state depends only on the previous state (not on older history) – Markov assumption
- Such system is called a Markov chain

# Markov assumption again

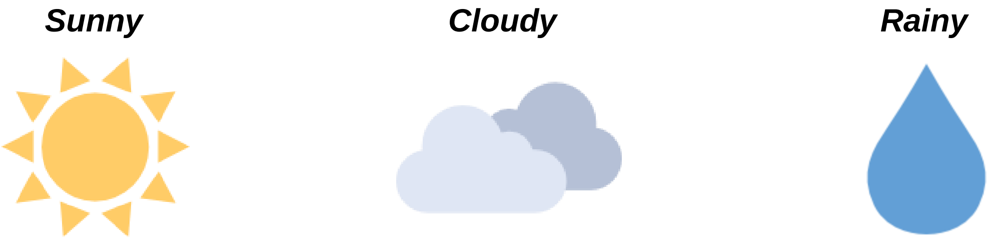
- The probability of any concrete state depends only on the previous state (not on older history):

$$P(\pi_i | \pi_1, \ldots, \pi_{i-1}) = P(\pi_i | \pi_{i-1})$$



# Markov chain example

- Weather prediction – predict tomorrow's weather
- Suppose that the weather can only be in 3 states:



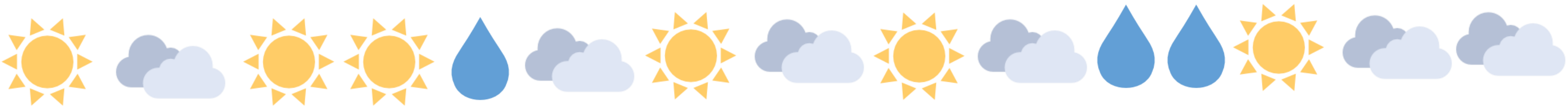
- Transition probability matrix: probability from state $\pi_{t-1}$ to state $\pi_t$:

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.4 | 0.3 | 0.3 |
| | Cloudy | 0.2 | 0.6 | 0.2 |
| | Sunny | 0.1 | 0.1 | 0.8 |

前一天的天气
affect 下一天

# Transition probability matrix

- Where does it come from?
  - Domain knowledge
  - Calculated from historical data, e.g. from the last 30 days:



- Consider all pairs of days and count the number of cases for all combinations of transitions, e.g. Rainy->Rainy, Rainy->Cloudy, Rainy->Sunny, etc.

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.4 | 0.3 | 0.3 |
| | Cloudy | 0.2 | 0.6 | 0.2 |
| | Sunny | 0.1 | 0.1 | 0.8 |

P(Rainy|Rainy) = 0.4
P(Cloudy|Rainy) = 0.3
P(Sunny|Rainy ) = 0.3

- The transition matrix includes conditional probabilities

# Let’s make some predictions!

- The previous 3 days are: Prediction for tomorrow?

- ☀ ☁ ☀ ☀ P(S|S) = 0.8

- 💧 ☁ ☀ ☀ P(S|S) = 0.8

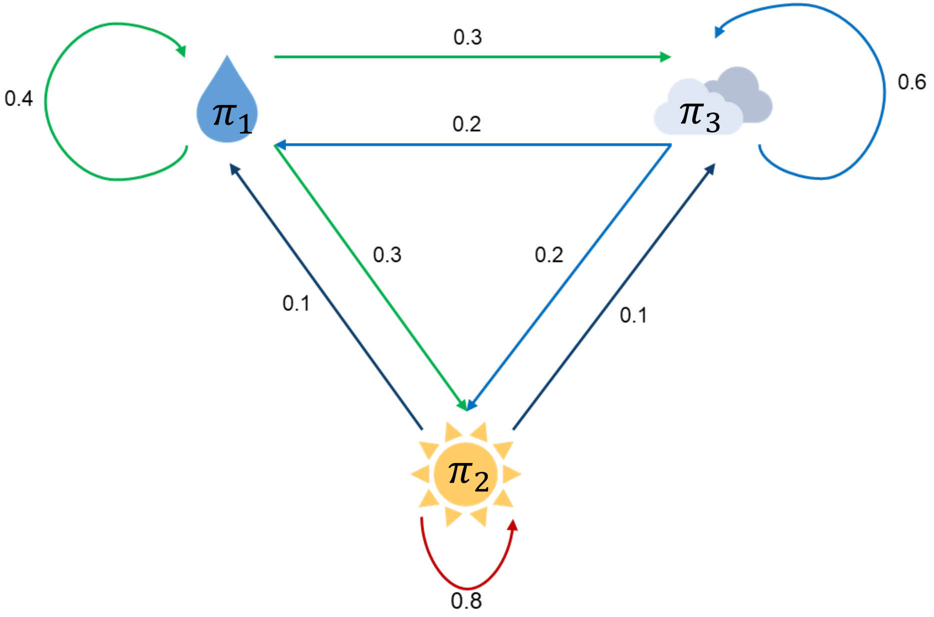
- ☀ 💧 ☁ ☁ P(C|C) = 0.6

- We only consider the last day (the current day) when we make the prediction for tomorrow. We use the transitional probability matrix and the max probability.

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.4 | 0.3 | 0.3 |
| | Cloudy | 0.2 | 0.6 | 0.2 |
| | Sunny | 0.1 | 0.1 | 0.8 |

# Markov chain diagram

- We can represent the Markov chain as a graph
  - The nodes correspond to the states
  - The links correspond to the transitions between the states
  - The weights on the links are the transition probabilities



- **Transition matrix** A
- **Transition probability** $\boldsymbol{a_{ij}}$ - the probability of moving from state i to state j

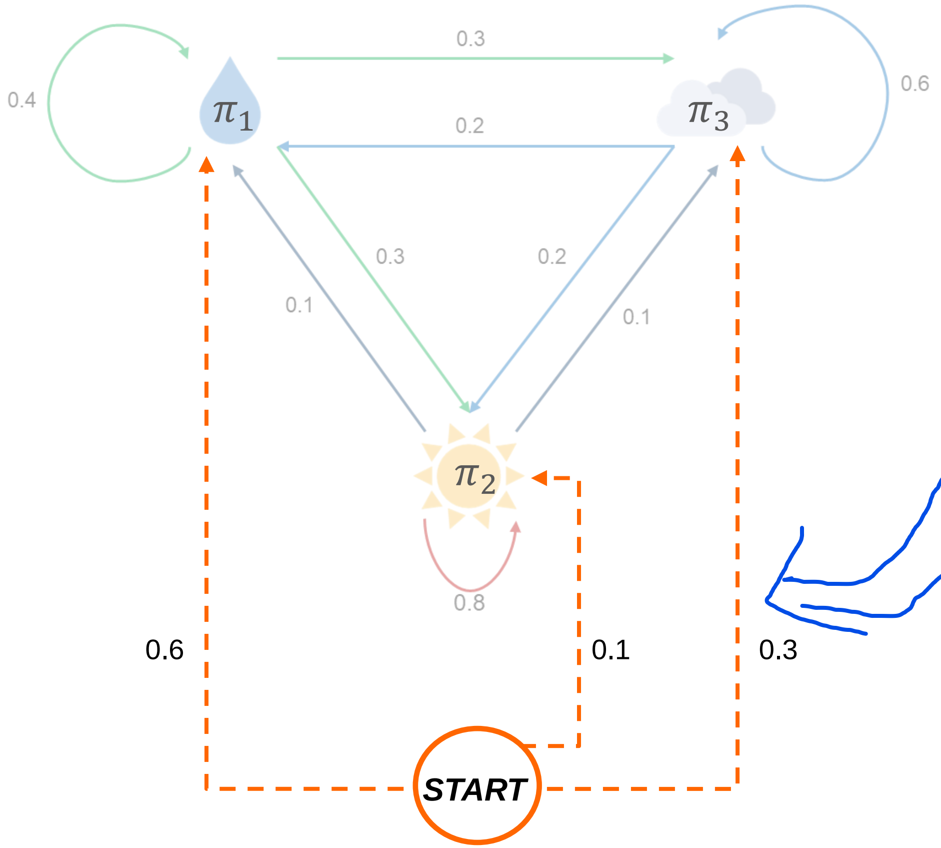
$$a_{ij} = P(\pi_t = j \mid \pi_{t-1} = i)$$

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \vdots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}$$

$\boldsymbol{\pi}$ – states
***A* -** state transition matrix

# Initial probabilities

- Adding a Start node and initial probabilities of the states



Transition probability matrix A

|  |  | $\pi_t$ |  |  |
|---|---|---|---|---|
|  |  | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.4 | 0.3 | 0.3 |
|  | Cloudy | 0.2 | 0.6 | 0.2 |
|  | Sunny | 0.1 | 0.1 | 0.8 |

Initial probability vector $A_0$

|  |  |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

- Initial probabilities of every state (vector)
- $A_0(\pi_t)$ - probability that the Markov chain will start in state $\pi_t$

$\boldsymbol{\pi}$ – states
***A*** – state transition matrix
$A_0$ – initial probabilities

# Predicting sequences

- We can use Markov chains to predict sequences of states, not only single states, e.g. to answer questions like this:
  - What will be the probability for sunny, sunny, cloudy, rainy?

- We can use the Markov rule:

$$P(\pi_1, \dots, \pi_k) = \overset{A_0}{P(\pi_1)} \prod_{i=1}^{k-1} P(\pi_{i+1} \mid \pi_i) = P(\pi_1)\, P(\pi_2 \mid \pi_1)\, P(\pi_3 \mid \pi_2) \cdots$$

  - It relates joint and conditional probabilities
  - It can be derived using the chain rule from statistics and the Markov assumption

# Predicting sequences - example

- What will be the probability for sunny, sunny, cloudy, rainy?

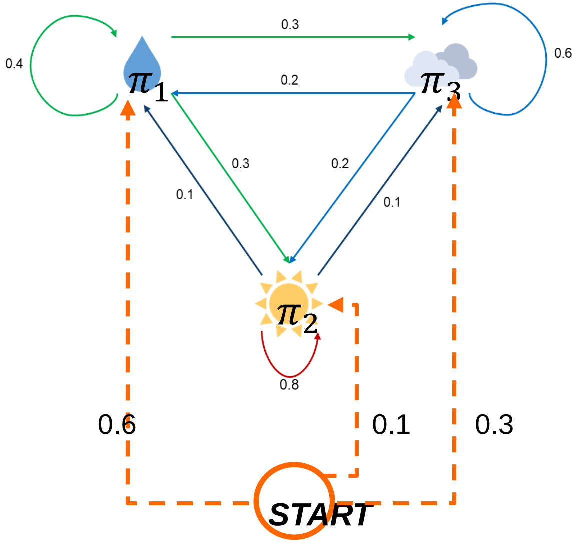
$$P(\pi_1, \ldots, \pi_k) = P(\pi_1) \prod_{i=1}^{k-1} P(\pi_{i+1} \,|\, \pi_i)$$

Transition probabilities

|  |  | $\pi_t$ |  |  |
|---|---|---|---|---|
|  |  | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.4 | 0.3 | 0.3 |
|  | Cloudy | 0.2 | 0.6 | 0.2 |
|  | Sunny | 0.1 | 0.1 | 0.8 |

Initial probabilities

|  |  |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |



- P(sunny, sunny, cloudy, rainy) =
=P(sunny)P(sunny|sunny)P(cloudy|sunny)P(rainy|cloudy)=
=0.1*0.8*0.1*0.2=0.0016

# Hidden Markov models

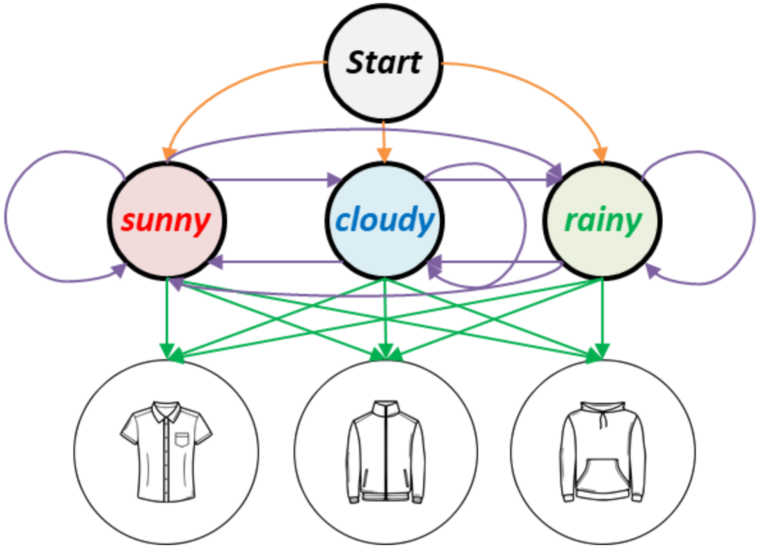
# Introduction

- Markov models are useful when we need to compute probability of directly observable states
- Hidden Markov models are used when we can’t observe the states directly (they are hidden) but we can develop judgement about them based on indirect observations

- A Hidden Markov model is a probabilistic model that allow us to **predict a sequence of hidden events** from a set of **observed events**

# Our weather example with hidden states

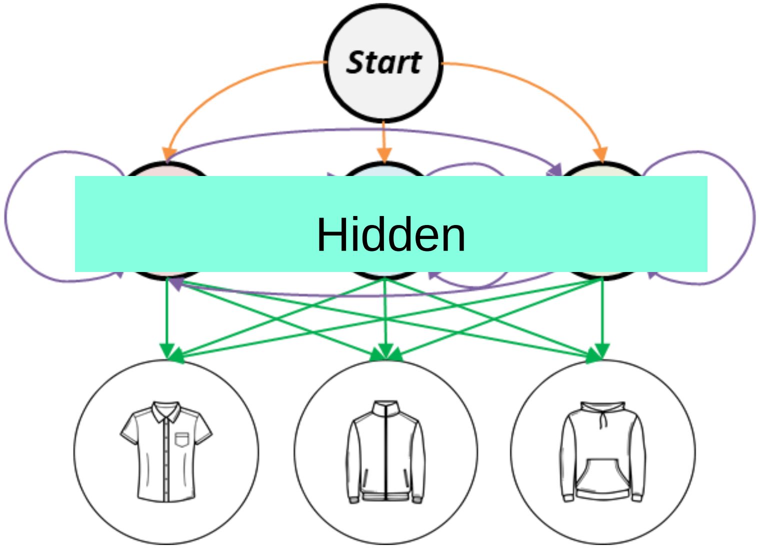
- You are locked in a room for several days. You don't know what is the weather outside. The only piece of evidence you have is what type of clothing the caregiver is wearing (the person who comes into the room to bring your meals). Can you guess what is the weather based on this indirect evidence?
  - What is hidden? The weather states: Sunny, Cloudy, Rainy
  - What can your see? Observations: Shirt, Jacket, Hoodie



| | |
|---|---|
| $\pi$ | states 👁 ***hidden*** |
| **X** | possible observations |
| **A** | state transition probabilities |
| **E** | emission probabilities |
| $A_0$ | initial probabilities |

# Our weather example with hidden states

- You are locked in a room for several days. You don’t know what is the weather outside. The only piece of evidence you have is what type of clothing the caregiver is wearing (the person who comes into the room to bring your meals). Can you guess what is the weather based on this indirect evidence?
  - What is hidden? The weather states: Sunny, Cloudy, Rainy
  - What can your see? Observations: Shirt, Jacket, Hoodie



| | |
|---|---|
| $\pi$ | states ***hidden*** |
| **X** | possible observations |
| **A** | state transition probabilities |
| **E** | emission probabilities |
| $A_0$ | initial probabilities |

# Another example of HMM

- John plays chess in his chess club every day
- He plays either against strong players (S), or against weak players (W). These are the states of HMM.
- Having played against an S for some time, he may prefer to choose a W and vice versa. These probabilities are stored in the transition matrix A.
- Based on whether he plays a S or W player, his chances of winning (+), loosing (-), or drawing (~) differ. These outcomes are the HMM observations. The probabilities of the observations are stored in the observation matrix X.
- When he comes home, he tells his mother the result of the game but not the strength of the opponent. Hence, his mother gets the observations.
- From the series of such observations during the week, and from the known A, X and $A_0$, she can calculate the probability that, say, the first 3 players were strong and the last 2 were weak.

Example from M. Kubat, An Introduction to Machine Learning, Third edition, Springer, 2021

# Components of HMM

N=3

- A set of *N* possible hidden states: $\pi = \pi_1, \dots, \pi_N$ sunny, cloudy, rainy
- A sequence of *M* observations: $X = x_1, x_2, \dots, x_M$ Shirt, Hoodie, Shirt, Jacket

M=4

- A transition probability matrix A, with probabilities $a_{ij}$ of moving from state *i* to state *j*

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

- An initial probability distribution over the states: $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

- An emission probability matrix E with elements $e_{\pi_i}(o_j)$, representing the conditional probabilities $P(o_j|\pi_i)$ of the observations for each state

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

- HMM model : $\lambda = (\pi, A, A_0)$ with observation sequence $X = x_1, x_2, \dots, x_M$

# The three main questions of HMM

- HMM problem 1 (also called Evaluation problem):
  - Given an HMM model $\lambda = (\pi, A, A_0)$ and an observation sequence $X = x_1, x_2$ , ..., $x_M$ , what is the probability of this sequence?

- HMM problem 2 (also called Decoding problem):
  - Given an HMM model $\lambda = (\pi, A, A_0)$ and an observation sequence $X = x_1, x_2$ , ..., $x_M$ , what is the most likely sequence of hidden states?

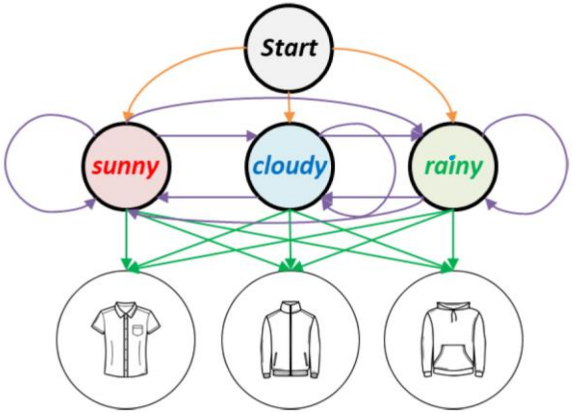
- HMM problem 3 (also called Learning problem):
  - Given an observation sequence $X = x_1, x_2$ , ..., $x_M$ , what is the model $\lambda = (\pi, A, A_0)$?

# HMM problem 1 (Evaluation)

# HMM problem 1 – weather example

- HMM problem 1 (also called Evaluation problem):
  - Given an HMM model $\lambda = (\pi, A, A_0)$ and an observation sequence $X = x_1, x_2$ , ..., $x_M$ , what is the probability of this observation sequence?

- Given the HMM model and the observation sequence *X= Shirt, Hoodie* (a sequence of clothing you observed in the past days), how likely is this sequence?



Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

# Solution by enumeration

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence X = Shirt, Hoodie   P(X)=?   Ex

- Solution by enumeration
- For the given observation sequence, compute the probability of each possible state sequence and sum these probabilities
- 9 sequences of states of length 2 ($3^2$):
  - Rainy, Rainy
  - Rainy, Cloudy
  - Rainy, Sunny
  - Cloudy, Rainy
  - Cloudy, Cloudy
  - Cloudy, Sunny
  - Sunny, Rainy
  - Sunny, Cloudy
  - Sunny, Sunny

P(X) = P(X, {Rainy, Rainy}) + P(X, {Rainy, Cloudy}) + P(X, {Rainy, Sunny} ) +
+ P(X, {Cloudy, Rainy}) + P(X, {Cloudy, Cloudy}) + P(X, {Cloudy, Sunny}) +
+ P (X, {Sunny, Rainy}) + P (X, {Sunny, Cloudy}) + P(X, {Sunny, Sunny})

# Solution by enumeration (2)

Method 1

Observation sequence X = Shirt, Hoodie   P(X)=?

P(X) = P(X, {Rainy, Rainy}) + P(X, {Rainy, Cloudy}) + P(X, {Rainy, Sunny} ) +
+ P(X, {Cloudy, Rainy}) + P(X, {Cloudy, Cloudy}) + P(X, {Cloudy, Sunny}) +
+ P (X, {Sunny, Rainy}) + P (X, {Sunny, Cloudy}) + P(X, {Sunny, Sunny})

T=2

N=3

- Each of these probabilities involves 4 multiplications, e.g.:

Assume

P(X, {Rainy Cloudy}) = P({Shirt Hoodie}, {Rainy Cloudy}) =
= P(Rainy)P(Cloudy|Rainy)P(Shirt|Rainy)P(Hoodie|Cloudy)

A   E

- So 9x4=36 multiplications for our simple example (N=3 states and a sequence of length *T*=2)
- The complexity is $2TN^T$ – *T* is the number of time steps (observation sequence length) and N is the number of states
- In practice, T is typically large and N is small -> inefficient algorithm
- This method will be slow for tasks with long sequences and many states!

# Forward algorithm - idea

method 2

- Instead, we can use the Forward algorithm, a dynamic programming algorithm, which stores a table of intermediate values (called forward probabilities) to make the computation more efficient
  - As before, we will sum over all possible hidden state paths but as we will keep the intermediate values, the computation will be faster
- Forward probability of state *k* at time *i:*

$$f_k(i) = e_k(x_i) \sum_j f_j(i-1)\, a_{jk}$$

This is the probability of the partial observation sequence up to time *i*

$$a_{jk} = \mathrm{P}(\pi_i = k | \pi_{i-1} = j)$$

$a_{jk}$*: transition probability from previous state* $\pi_j$ *to current state* $\pi_k$

$$e_k(x_i) = P(x_i | \pi_i = k)$$

$e_k(x_i)$*: emission probability of observation* $x_i$ *given the current state k*

- Key insight: $f_k(i)$ can be computed recursively based on $f_j(i-1)$
- Efficient algorithm: complexity is $N^2T$ – better than $2TN^T$ (T is large, N is small)

Method 2 Method 1

# Forward algorithm

- Step 1: Initialization: $f_k(1) = \overset{\text{initial}}{\underline{A_0(k)}}e_k(x_1)$
  - For time step *t* =1, compute:
  - $f_k(1)$ - forward probability of state *k* at time step 1
  - $x_1$- observation a time *t* =1

- Step 2: Iteration:
- For time steps *t* = 2,..,*m*, compute:

$$f_k(i) = e_k(x_i)\sum_j f_j(i-1)\,a_{jk}$$

(forward probability of state *k* at time step *i )*

$a_{jk}$*: transition probability from previous state* $\pi_j$ *to current state* $\pi_k$
$e_k(x_i)$*: emission probability of observation* $x_i$ *given the current state k*

$$a_{jk}\text{=P}(\pi_i = k|\pi_{i-1} = j)$$

$$e_k(x_i) = P(x_i|\pi_i = k)$$

- Step 3: Termination

$$P(x) = \sum_k f_k(m)$$

*m* is the last time step

# Back to our example, solution using the Forward algorithm – step 1

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence: X = Shirt, Hoodie

## Forward algorithm

- Step 1: Initialization: $f_k(1) = A_0(k)e_k(x_1)$
  - $f_k(1)$ - forward probability of state *k* at time *1*
  - $x_1$ = *Shirt*
  - 3 states: *Rainy, Cloudy, Sunny*

$f_{Rainy}(1) = A_0(Rainy)\, e_{Rainy}(Shirt)$ = 0.6*0.8=0.48

$f_{Cloudy}(1) = A_0(Cloudy)\, e_{Cloudy}(Shirt)$ = 0.3*0.5=0.15

$f_{Sunny}(1) = A_0(Sunny)\, e_{Sunny}(Shirt) =$ 0.1*0.01=0.001

# Solution using the Forward algorithm – step 2

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence: X = Shirt, Hoodie

- Step 2: Iteration

$f_k(i)$ - forward probability of state *k* at time *i:*

$$\underline{f_k(i) = e_k(x_i) \sum_j f_j(i-1)\, a_{jk}}$$

- time step *t* = 2, $x_2$ = *Hoodie*
- 3 states: *Rainy, Cloudy, Sunny*

- 1) For *Rainy:*

$$\boxed{f_{Rainy}(2)} =$$

$$= e_{Rainy}(Hoodie)\left(f_{Rainy}(1)a_{RainyRainy} + f_{Cloudy}(1)a_{CloudyRainy} + f_{Sunny}(1)a_{SunnyRainy}\right)$$

$$= 0.01*(0.48*0.6+0.15*0.4+0.001*0.1)=0.01*0.3481=0.0035$$

# Solution using the Forward algorithm – step 2 (cont.)

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence: X = Shirt, Hoodie

- 2) For *Cloudy:*

$$f_{Cloudy}(2) =$$

$$= e_{Cloudy}(Hoodie)\left(f_{Rainy}(1)a_{RainyCloudy} + f_{Cloudy}(1)a_{CloudyCloudy} + f_{Sunny}(1)a_{SunnyCloudy}\right)$$

$$= 0.1*(0.48*0.3+0.15*0.3+0.001*0.4)=0.1*0.1894=0.0189$$

- 3) For *Sunny:*

$$f_{Sunny}(2) =$$

$$= e_{Sunny}(Hoodie)\left(f_{Rainy}(1)a_{RainySunny} + f_{Cloudy}(1)a_{CloudySunny} + f_{Sunny}(1)a_{SunnySunny}\right)$$

$$= 0.79*(0.48*0.1+0.15*0.3+0.001*0.5) = 0.0739$$

# Solution using the Forward algorithm – step 3

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence: X = Shirt, Hoodie

- Step 3: Termination
  - $f_k(m)$ - forward probability of state *k* at time *m* – at the last time step
  - *m* = 2

$$P(X) = \sum_k f_k(m)$$

$$P(X) = P(Shirt, Hoodie) \underline{= f_{Rainy}(2) + f_{Cloudy}(2) + f_{Sunny}(2) =}$$

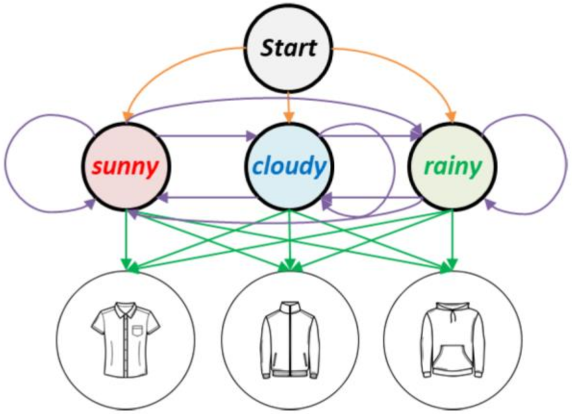
$$= \ 0.0035 + 0.0189 + 0.0739 = 0.0963$$

# HMM problem 2 (Decoding)

# HMM problem 2 – weather example

- HMM problem 1 (also called Decoding problem):
  - Given an HMM model $\lambda = (\pi, A, A_0)$ and an observation sequence $X = x_1, x_2$ , ..., $x_M$ , what is the , what is the most likely sequence of hidden states?

- Given the HMM model and the observation sequence *X= Shirt, Hoodie,* what is the most likely sequence of hidden states?



Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

# Viterbi algorithm - idea

- Instead, we can use the Viterbi algorithm, a dynamic programming algorithm, which calculates the probability of the best path ending at each of the states (called Viterbi score)
- The Viterbi algorithm is efficient because at each stage, it only needs to maintain the highest scoring path ending at each state, not a list of all possible paths
- Viterbi score of state *k* at time *i:*

$$V_k(i) = e_k(x_i) \max_j V_j(i-1)\ a_{jk}$$

Probability of most likely sequence of states ending at state $\pi_i = k$

$$a_{jk} = \mathrm{P}(\pi_i = k | \pi_{i-1} = j)$$

*$a_{jk}$: transition probability from previous state $\pi_j$ to current state $\pi_k$*

$$e_k(x_i) = P(x_i | \pi_i = k)$$

*$e_k(x_i)$: emission probability of observation $x_i$ given the current state k*

- Key insight: $V_k(i)$ can be computer recursively based on $V_j(i-1)$

# Viterbi algorithm – idea (2)

- However, the Viterbi score gives the probability of best path ending with the final state. We can determine the final state but not the path itself.
- To determine the path, we need to trace-back from the final state
- To do this, we keep a back-pointer of each state when we calculate the Viterbi score:

$$Ptr_k(i) = \underset{j}{\operatorname{argmax}} V_j(i-1)a_{jk}$$

# Viterbi algorithm

- Step 1: Initialization: $V_k(1) = A_0(k)e_k(x_1)$
  - For time step *t* =1, compute:
  - $V_k(1)$ - Viterbi score of state *k* at time step 1
- Step 2: Iteration:
- For time steps *t* =2,..,*m*, compute:

  1) Viterbi score of state *k* at time *i*:

  $$V_k(i) = e_k(x_i)\max_j V_j(i-1)a_{jk}$$

  2) Back-pointer of state *k* at time *i*

  $$Ptr_k(i) = \operatorname*{argmax}_j V_j(i-1)a_{jk}$$

*$a_{jk}$: transition probability from previous state $\pi_j$ to current state $\pi_k$*

*$e_k(x_i)$: emission probability of observation $x_i$ given the current state k*

$$a_{jk} = \mathrm{P}(\pi_i = k|\pi_{i-1} = j)$$

$$e_k(x_i) = P(x_i|\pi_i = k)$$

- Step 3: Termination and trace-back
- 1) Determine the final state (at the last time step *m*):

  $$\boxed{\pi_m} = \operatorname*{argmax}_k \underline{V_k}(m)$$

- 2) Trace-back the previous states:

  $$\pi_{i-1} = Ptr_{\pi_i}(i) \quad i = m,..,,2$$

$\pi_m \rightarrow \pi_{m-1} \rightarrow \pi_{m-2} \rightarrow \pi_{m-3}$ ----

# Back to our example, solution using the Viterbi algorithm – step 1

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence X = Shirt, Hoodie

## Viterbi algorithm

- Step 1: Initialization: $V_k(1) = A_0(k)e_k(x_1)$
  - $V_k(1)$ - Viterbi score of state *k* at time *1*
  - $x_1$ = *Shirt*
  - 3 states: *Rainy, Cloudy, Sunny*

$V_{Rainy}(1) = A_0(Rainy)\, e_{Rainy}(Shirt) =$ 0.6*0.8=0.48

$V_{Cloudy}(1) = A_0(Cloudy)\, e_{Cloudy}(Shirt) =$ 0.3*0.5=0.15

$V_{Sunny}(1) = A_0(Sunny)\, e_{Sunny}(Shirt) =$ 0.1*0.01=0.001

# Solution using the Viterbi algorithm – step 2 (1)

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence = Shirt, Hoodie

- Step 2: Iteration
  - $V_k(i)$ - Viterbi score of state *k* at time *i*

    $$V_k(i) = e_k(x_i) \max_j V_j(i-1)a_{jk}$$

  - time step *t* = 2, $x_2$ = *Hoodie*

  - *1) For Rainy*

$$V_{Rainy}(2) = e_{Rainy}(Hoodie) * \max\left(V_{Rainy}(1)a_{RainyRainy}, \quad V_{Cloudy}(1)a_{CloudyRainy}, \quad V_{Sunny}(1)a_{SunnyRainy}\right)$$

$$= 0.01*\max(\mathbf{0.48*0.6,}\ 0.15*0.4,\ 0.001*0.1) = 0.01*0.48*0.6 = 0.0029$$

$$Ptr_{Rainy}(2) = \text{argmax}(\mathbf{0.48*0.6,}\ 0.15*0.4,\ 0.001*0.1) = 1, \text{ i.e. Rainy}$$

# Solution using the Viterbi algorithm – step 2 (1)

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence = Shirt, Hoodie

- Step 2: Iteration
  - $V_k(i)$ - Viterbi score of state *k* at time *i*

    $$V_k(i) = e_k(x_i) \max_j V_j(i-1)a_{jk}$$
  - time step *t* = 2, $x_2$ = *Hoodie*

  - *1) For Rainy*

$$V_{Rainy}(2) = e_{Rainy}(Hoodie) * \max\left(\overset{\text{j= Rainy}}{V_{Rainy}(1)a_{RainyRainy}}, \overset{\text{j= Cloudy}}{V_{Cloudy}(1)a_{CloudyRainy}}, \overset{\text{j= Sunny}}{V_{Sunny}(1)a_{SunnyRainy}}\right)$$

$$= 0.01*\max(\mathbf{0.48*0.6,}\ 0.15*0.4,\ 0.001*0.1)=0.01*0.48*0.6=0.0029$$

$$Ptr_{Rainy}(2) = \text{argmax}(\mathbf{0.48*0.6,}\ 0.15*0.4,\ 0.001*0.1) = 1, \text{ i.e. Rainy}$$

# Solution using the Viterbi algorithm – step 2 (2)

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence = Shirt, Hoodie

- Step 2: Iteration
  - $V_k(i)$ - Viterbi score of state *k* at time *i*

    $$V_k(i) = e_k(x_i) \max_j V_j(i-1)a_{jk}$$
  - time step *t* = 2, $x_2$ = *Hoodie*

  - *2) For Cloudy*

$$V_{Cloudy}(2) = e_{Cloudy}(Hoodie) \ * $$

$$\max\left(V_{Rainy}(1)a_{RainyCloudy}, V_{Cloudy}(1)a_{CloudyCloudy}, V_{Sunny}(1)a_{SunnyCloudy}\right)$$

$$= 0.1*(\mathbf{0.48*0.3,}\ 0.15*0.3,\ 0.001*0.4)=0.1*0.48*0.3=0.0144$$

$$Ptr_{Cloudy}(2) \ = \text{argmax}(\mathbf{0.48*0.3,} 0.15*0.3, 0.001*0.4) = 1, \text{i.e. Rainy}$$

# Solution using the ~~Viterbi algorithm –~~ step 2 (3)

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence = Shirt, Hoodie

- Step 2: Iteration
  - $V_k(i)$ - Viterbi score of state *k* at time *i*

    $$V_k(i) = e_k(x_i) \max_j V_j(i-1) a_{jk}$$
  - time step *t* = 2, $x_2$ = *Hoodie*

  - *3) For Sunny*

$$V_{Sunny}(2) = e_{Sunny}(Hoodie) * \max\left(V_{Rainy}(1) a_{RainySunny}, V_{Cloudy}(1) a_{CloudySunny}, V_{Sunny}(1) a_{SunnySunny}\right)$$

$$= 0.79 * (\mathbf{0.48 * 0.1,}\ 0.15*0.3,\ 0.001*0.5) = 0.79*0.48*0.1 = 0.0379$$

$$Ptr_{Sunny}(2) = \text{argmax}(\mathbf{0.48 * 0.1,}\ 0.15*0.3, 0.001*0.5) = 1, \text{i.e. Rainy}$$

# Solution using the Viterbi algorithm – step 3

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Observation sequence = Shirt, Hoodie

- Step 3: Termination and trace-back

$$\pi_N = \underset{k}{\operatorname{argmax}} V_k(N)$$

The final state (it is for time t=2) is given by:

$\operatorname{argmax}\left(V_{Rainy}(2), V_{Cloudy}(2), V_{Sunny}(2)\right) =$ ~~$\operatorname{argmax}\left(V_{Rainy}(2), V_{Cloudy}(2), V_{Sunny}(2)\right)$~~ $=$

$= \operatorname{argmax}(0.0029, 0.0144, \mathbf{0.0379}) = 3$, i.e. Sunny

Backtracking through the pointers:

$$Ptr_{Sunny}(2) = \text{Rainy}$$

Hence, the most likely sequence of hidden states is Rainy, Sunny

# Forward and Viterbi algorithms - comparison

## Forward

**Step 1: Initialization:**

$$f_k(1) = A_0(k)e_k(x_1)$$

**Step 2: Iteration:**

$$f_k(i) = e_k(x_i)\sum_j f_j(i-1)\, a_{jk}$$

**Step 3: Termination:**

$$P(X) = \sum_k f_k(m)$$

## Viterbi

**Step 1: Initialization:**

$$V_k(1) = A_0(k)e_k(x_1)$$

**Step 2: Iteration:**

$$V_k(i) = e_k(x_i)\max_j V_j(i-1)a_{jk}$$

**Step 3: Termination:**

$$\pi_m = \operatorname*{argmax}_k V_k(m)$$

Sequence →

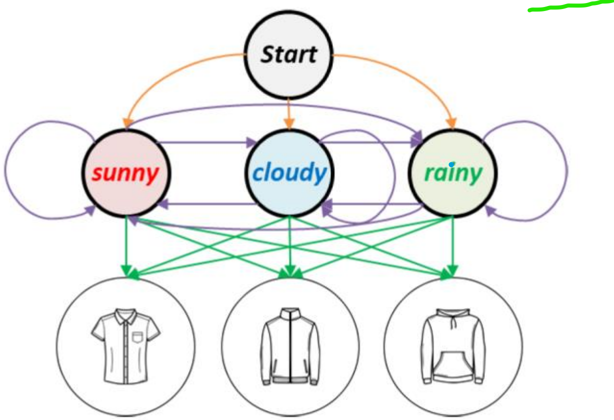
$$\pi_{i-1} = Ptr_{\pi_i}(i)$$

# HMM problem 3 (Learning)

# HMM problem 3 – weather example

- HMM problem 3 (also called Learning problem):
- Given an observation sequence $X = x_1, x_2$ , ..., $x_M$ , what is the model $\lambda = (\pi, A, A_0)$?

- Given the observation sequence *X= Shirt, Hoodie* (a sequence of clothing you observed in the past days), what is the model $\lambda = (\pi, A, A_0)$?



Transition probability matrix A

| | | $\pi_t$ | | |
|---|---|---|---|---|
| | | Rainy | Cloudy | Sunny |
| $\pi_{t-1}$ | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

Initial probabilities of states $A_0$

| | |
|---|---|
| Rainy | 0.6 |
| Cloudy | 0.3 |
| Sunny | 0.1 |

Emission probability matrix E

| | Shirt | Jacket | Hoodie |
|---|---|---|---|
| Rainy | 0.8 | 0.19 | 0.01 |
| Cloudy | 0.5 | 0.4 | 0.1 |
| Sunny | 0.01 | 0.2 | 0.79 |

Find them!

# HMM problem 3 – solution outline

- Uses the Expectation Maximization (EM) algorithm – outline:

1. 1. Initialize $\lambda = (\pi, A, A_0)$ to random values
2. 2. Compute the probabilities $A, \text{and } A_0$ for all time steps, for all states $\pi$ (expectation step)
3. 3. Re-estimate $\lambda = (\pi, A, A_0)$ based on the observation sequence $X$ – use the probabilities to predict the most likely observation sequence and compare with the actual observation sequence (maximization step)
4. If P(X| $\lambda$ ) have increased, return to step 2, otherwise stop.

# Applications of HMM

- In speech processing: Based on a sequence of words, identify the underlying sequence of Part-of-Speech (POS) tags
  - POS - noun, verb, adjective, adverb, etc.
- HMM formulation:
  - States: noun, adjective, …. (the POS tags)
  - Observations: concrete words
- Task: construct HMM $\lambda = (\pi, A, A_0)$ which will predict the tag for observed words in the sequence
- Why is it possible? Because words and tags are related
- Applications of POS tagging: word sense disambiguation, named entity recognition, sentiment analysis, question answering
- Example: word pronunciation – words are often pronounced differently based on their POS (e.g. adjective or verb - “approximate” )

# Applications of HMM (2)

- In finance
  - Predict the sequences of events from observed prices of shares
- In bioinformatics
  - Predict the behavior of RNA nucleotides
- In computer vision:
  - Recover the original event from an observed sequence
  - Gesture recognition
- In information security
  - Predict the behavior of malware from a sequence of observations

# Summary

- A Markov chain (Markov process) is a model describing a sequence of transitions from one state to another, in which the probability of each state depends only on the previous state
- Hidden Markov Models (HMM) are described by a set of states and a set of observations
- In HMM we don’t have a direct access to the states (they are hidden), only to the observations from which the probabilities of the states can be obtained
- There are three fundamental problems of HMMs:
  - HMM problem 1: Find the probability of the observation sequence – solved with the Forward algorithm
  - HMM problem 2: Find the most likely hidden state sequence – solved with the Viterbi algorithm
  - HMM problem 3: Find the HMM model – solved with the EM algorithm

# Summary (2)

- HMMs are powerful, efficient and very useful in practice
- They do not make assumptions about the data
- They can extract a lot of statistical information from the data
- They have proven useful in many applications, especially for speech processing, natural language processing and information security

- Extensions of HMM
  - Maximum Entropy Markov Models
  - Conditional Random Fields

- => When a temporal pattern is hidden behind indirect observations, we can use HMMs