

Markov Models

COMP5318 Machine Learning and Data Mining
semester 1, 2023, week 11

Irena Koprinska

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



THE UNIVERSITY OF
SYDNEY



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



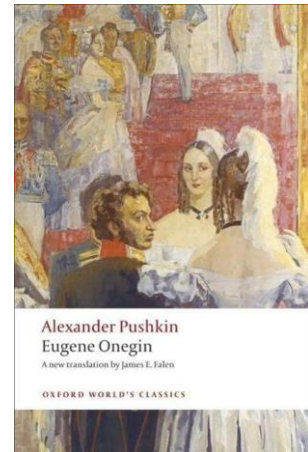
Markov models

- 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:

$$P = \begin{matrix} & \begin{matrix} \text{vowel} & \text{consonant} \end{matrix} \\ \begin{matrix} \text{vowel} \\ \text{consonant} \end{matrix} & \begin{pmatrix} 0.175 & 0.825 \\ 0.526 & 0.474 \end{pmatrix} \end{matrix}$$



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

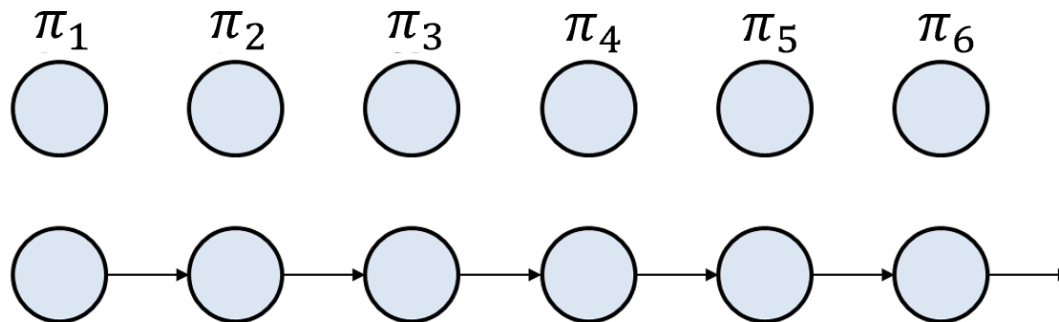


<https://www.amazon.com.au/Eugene-Onegin-Novel-Alexander-Pushkin/dp/0199538646>

- 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, \dots
 - At each time, t_i , the system is in a state π_i , e.g. $\pi_1 = \text{hot}$, then $\pi_2 = \text{cold}$, etc.
 - Between two subsequent time steps t_i and t_{i+1} , the system changes its state from π_t to π_{t+1} . The transition from π_t to π_{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**

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

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



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



- **Transition probability matrix:** probability from state π_{t-1} to state π_t :

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{t-1} | Rainy | 0.4 | 0.3 | 0.3 |
| | Cloudy | 0.2 | 0.6 | 0.2 |
| | Sunny | 0.1 | 0.1 | 0.8 |

- 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.

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{t-1} | Rainy | 0.4 | 0.3 | 0.3 |
| | Cloudy | 0.2 | 0.6 | 0.2 |
| | Sunny | 0.1 | 0.1 | 0.8 |

$$P(\text{Rainy}|\text{Rainy}) = 0.4$$

$$P(\text{Cloudy}|\text{Rainy}) = 0.3$$

$$P(\text{Sunny}|\text{Rainy}) = 0.3$$

- The transition matrix includes conditional probabilities

Let's make some predictions!

- The previous 3 days are:



•



•



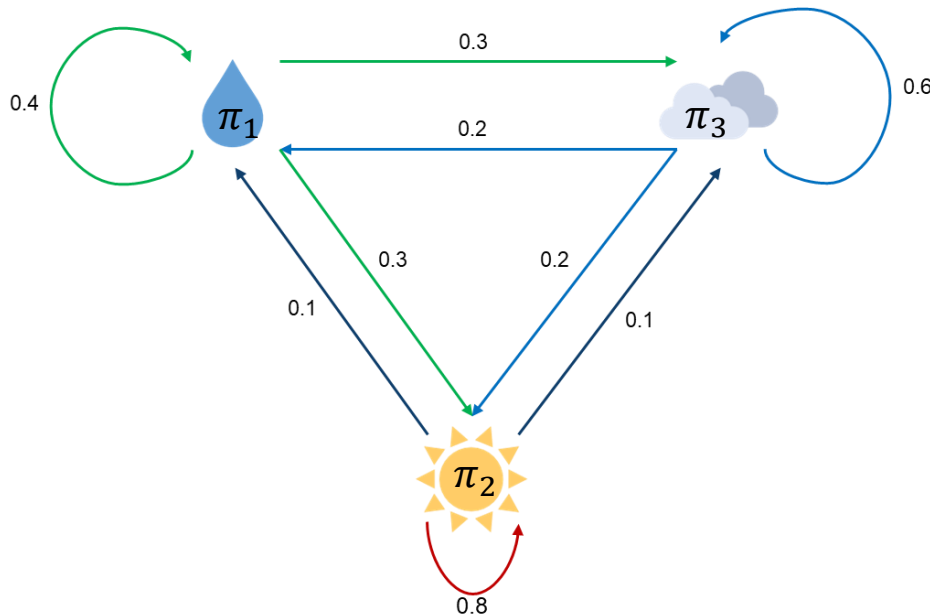
Prediction for tomorrow?



- 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.

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{t-1} | Rainy | 0.4 | 0.3 | 0.3 |
| | Cloudy | 0.2 | 0.6 | 0.2 |
| | Sunny | 0.1 | 0.1 | 0.8 |

- 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** a_{ij} - the probability of moving from state i to state j

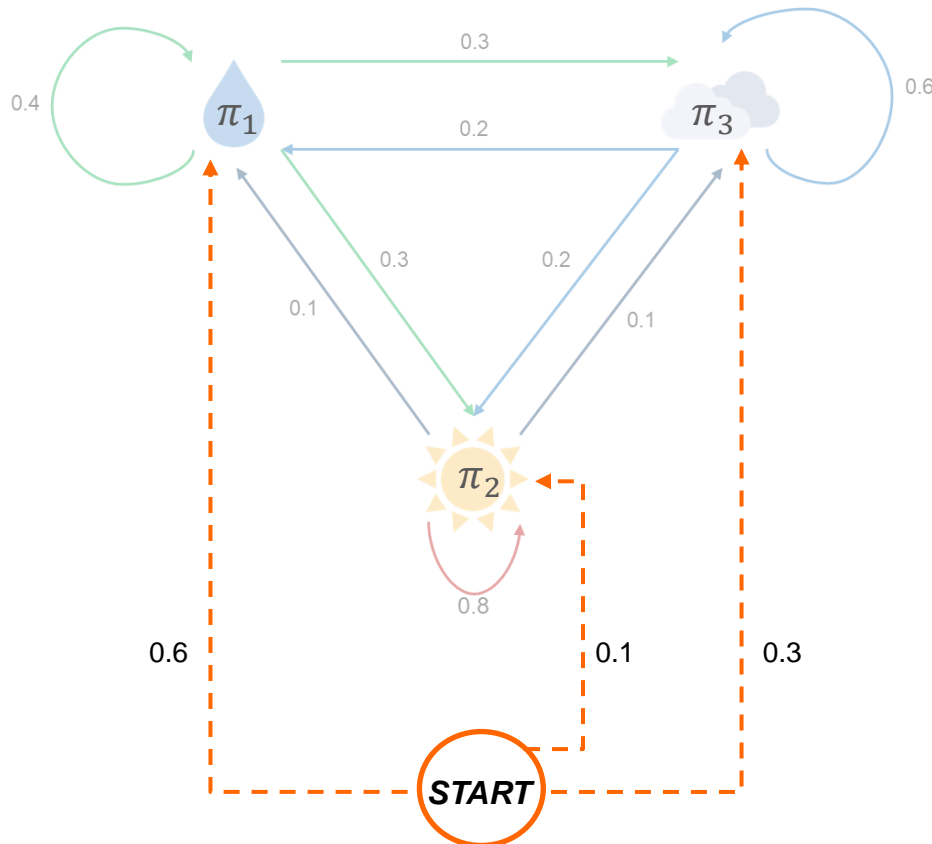
$$a_{ij} = P(\pi_t = j | \pi_{t-1} = i)$$

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

π – states

A - state transition matrix

- Adding a Start node and **initial probabilities of the states**



Transition probability matrix A

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 π_t

π – states

A – state transition matrix

A_0 – **initial probabilities**

- 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:

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

- 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?

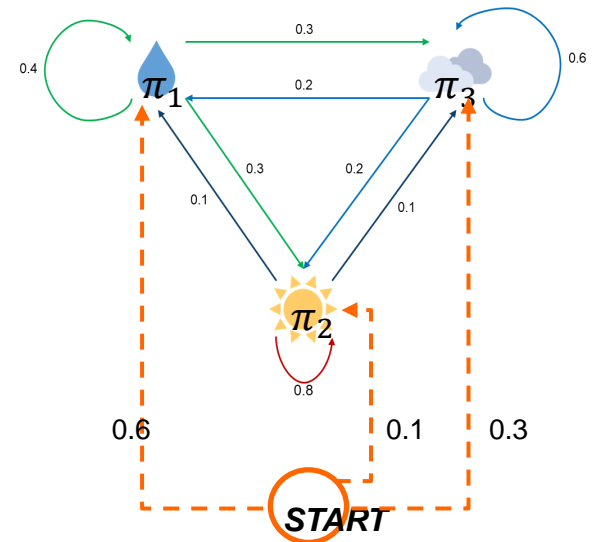
$$P(\pi_1, \dots, \pi_k) = P(\pi_1) \prod_{i=1}^{k-1} P(\pi_{i+1} | \pi_i)$$

Transition probabilities

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 |



- $P(\text{sunny, sunny, cloudy, rainy}) =$
 $= P(\text{sunny})P(\text{sunny}|\text{sunny})P(\text{cloudy}|\text{sunny})P(\text{rainy}|\text{cloudy}) =$
 $= 0.1 * 0.8 * 0.1 * 0.2 = 0.0016$

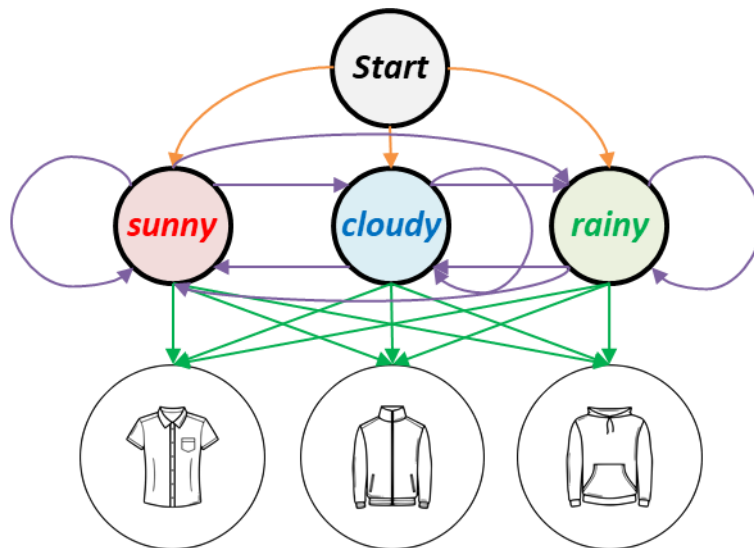



Hidden Markov models

- 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

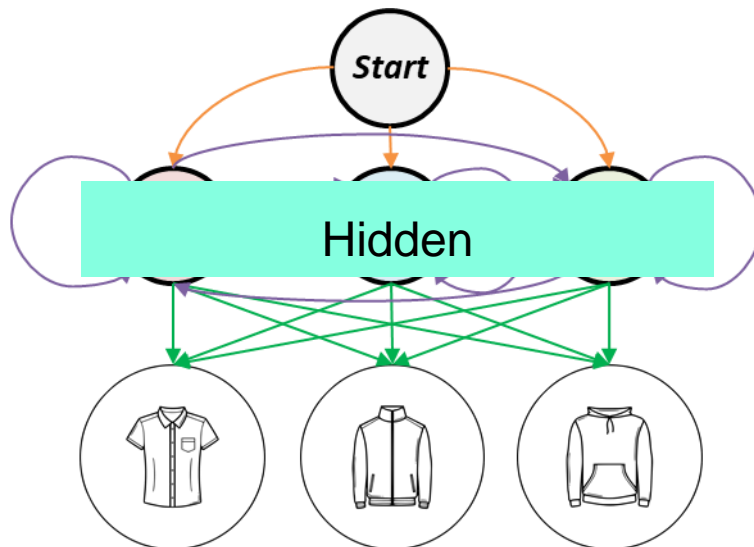
- You are locked in a room for several days. You don't know what is the weather outside. The only piece of evidence you have is what type of clothing the caregiver is wearing (the person who comes into the room to bring your meals). Can you guess what is the weather based on this indirect evidence?
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




π 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 you see? Observations: Shirt, Jacket, Hoodie



π states  hidden
 X possible observations
 A state transition probabilities
 E emission probabilities
 A_0 initial probabilities

- 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

Irena Koprinska, irena.koprinska@sydney.edu.au COMP5318 ML&DM, week 11, 2023

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{t-1} | Rainy | 0.6 | 0.3 | 0.1 |
| | Cloudy | 0.4 | 0.3 | 0.3 |
| | Sunny | 0.1 | 0.4 | 0.5 |

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

- An initial probability distribution over the states: A_0
- 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

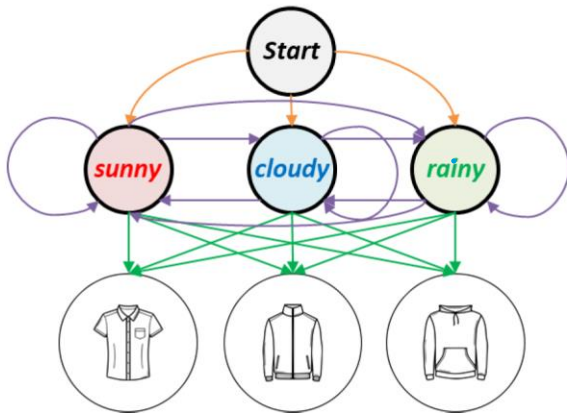
- HMM problem 1 (also called Evaluation problem):
 - Given an HMM model $\lambda = (\pi, A, A_0)$ and an observation sequence $X = x_1, x_2, \dots, x_M$, what is the probability of this sequence?
- HMM problem 2 (also called Decoding problem):
 - Given an HMM model $\lambda = (\pi, A, A_0)$ and an observation sequence $X = x_1, x_2, \dots, x_M$, what is the most likely sequence of hidden states?
- HMM problem 3 (also called Learning problem):
 - Given an observation sequence $X = x_1, x_2, \dots, x_M$, 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 $X = x_1, x_2, \dots, x_M$, what is the probability of this observation sequence?
- Given the HMM model and the observation sequence $X = \text{Shirt}, \text{Hoodie}$ (a sequence of clothing you observed in the past days), **how likely is this sequence?**



Transition probability matrix A

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 |

Initial probabilities of states A_0

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

Transition probability matrix A

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 $X = \text{Shirt, Hoodie}$ $P(X)=?$

- 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 Cloudy, Rainy Sunny, Rainy
 - Rainy, Cloudy Cloudy, Cloudy Sunny, Cloudy
 - Rainy, Sunny Cloudy, Sunny Sunny, Sunny

$$\begin{aligned}
 P(X) = & P(X, \{\text{Rainy, Rainy}\}) + P(X, \{\text{Rainy, Cloudy}\}) + P(X, \{\text{Rainy, Sunny}\}) + \\
 & + P(X, \{\text{Cloudy, Rainy}\}) + P(X, \{\text{Cloudy, Cloudy}\}) + P(X, \{\text{Cloudy, Sunny}\}) + \\
 & + P(X, \{\text{Sunny, Rainy}\}) + P(X, \{\text{Sunny, Cloudy}\}) + P(X, \{\text{Sunny, Sunny}\})
 \end{aligned}$$

Observation sequence $X = \text{Shirt, Hoodie}$ $P(X)=?$

$$\begin{aligned} P(X) = & P(X, \{\text{Rainy, Rainy}\}) + P(X, \{\text{Rainy, Cloudy}\}) + P(X, \{\text{Rainy, Sunny}\}) + \\ & + P(X, \{\text{Cloudy, Rainy}\}) + P(X, \{\text{Cloudy, Cloudy}\}) + P(X, \{\text{Cloudy, Sunny}\}) + \\ & + P(X, \{\text{Sunny, Rainy}\}) + P(X, \{\text{Sunny, Cloudy}\}) + P(X, \{\text{Sunny, Sunny}\}) \end{aligned}$$

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

$$\begin{aligned} P(X, \{\text{Rainy Cloudy}\}) &= P(\{\text{Shirt Hoodie}\}, \{\text{Rainy Cloudy}\}) = \\ &= P(\text{Rainy})P(\text{Cloudy}|\text{Rainy})P(\text{Shirt}|\text{Rainy})P(\text{Hoodie}|\text{Cloudy}) \end{aligned}$$

- So $9 \times 4 = 36$ 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 \rightarrow inefficient algorithm
- This method will be **slow** for tasks with long sequences and many states!

- 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

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

a_{jk} : transition probability from previous state π_j to current state π_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)

- **Step 1: Initialization:** $f_k(1) = A_0(k)e_k(x_1)$

- For time step $t=1$, compute:
- $f_k(1)$ - forward probability of state k at time step 1
- x_1 - observation at time $t=1$

- **Step 2: Iteration:**

- For time steps $t = 2, \dots, m$, 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 π_j to current state π_k

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

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

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

- **Step 3: Termination**

$$P(x) = \sum_k f_k(m) \quad m \text{ is the last time step}$$

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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: $X = \text{Shirt, Hoodie}$

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
 - $x_1 = \text{Shirt}$
 - 3 states: *Rainy, Cloudy, Sunny*

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

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

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

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 :
$$f_k(i) = e_k(x_i) \sum_j f_j(i-1) a_{jk}$$

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

- 1) For *Rainy*:

$$\begin{aligned} f_{\text{Rainy}}(2) &= \\ &= e_{\text{Rainy}}(\text{Hoodie}) (f_{\text{Rainy}}(1) a_{\text{RainyRainy}} + f_{\text{Cloudy}}(1) a_{\text{CloudyRainy}} + f_{\text{Sunny}}(1) a_{\text{SunnyRainy}}) \\ &= 0.01 * (0.48 * 0.6 + 0.15 * 0.4 + 0.001 * 0.1) = 0.01 * 0.3481 = 0.0035 \end{aligned}$$

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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*:

$$\begin{aligned}
 f_{\text{Cloudy}}(2) &= \\
 &= e_{\text{Cloudy}}(\text{Hoodie}) (f_{\text{Rainy}}(1)a_{\text{RainyCloudy}} + f_{\text{Cloudy}}(1)a_{\text{CloudyCloudy}} + f_{\text{Sunny}}(1)a_{\text{SunnyCloudy}}) \\
 &= 0.1*(0.48*0.3+0.15*0.3+0.001*0.4)=0.1*0.1894=0.0189
 \end{aligned}$$

- 3) For *Sunny*:

$$\begin{aligned}
 f_{\text{Sunny}}(2) &= \\
 &= e_{\text{Sunny}}(\text{Hoodie}) (f_{\text{Rainy}}(1)a_{\text{RainySunny}} + f_{\text{Cloudy}}(1)a_{\text{CloudySunny}} + f_{\text{Sunny}}(1)a_{\text{SunnySunny}}) \\
 &= 0.79*(0.48*0.1+0.15*0.3+0.001*0.5) = 0.0739
 \end{aligned}$$

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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: $X = \text{Shirt, Hoodie}$

• Step 3: Termination

- $f_k(m)$ - forward probability of state k at time m – at the last time step
- $m = 2$

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

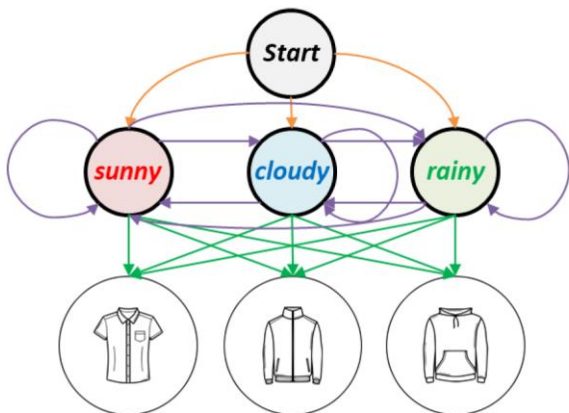
$$P(X) = P(\text{Shirt, Hoodie}) = f_{\text{Rainy}}(2) + f_{\text{Cloudy}}(2) + f_{\text{Sunny}}(2) = \\ = 0.0035 + 0.0189 + 0.0739 = 0.0963$$



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 $X = x_1, x_2, \dots, x_M$, what is the most likely sequence of hidden states?
- Given the HMM model and the observation sequence $X = \text{Shirt}, \text{Hoodie}$, what is the most likely sequence of hidden states?



Transition probability matrix A

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 |

- 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 $\pi_i = k$

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

a_{jk} : transition probability from previous state π_j to current state π_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 computed recursively based on $V_j(i-1)$

- 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:

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

- **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 $t=2, \dots, m$, 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

$$Ptr_k(i) = \operatorname{argmax}_j V_j(i-1)a_{jk}$$
- **Step 3: Termination and trace-back**
 - 1) Determine the final state (at the last time step m):

$$\pi_m = \operatorname{argmax}_k V_k(m)$$
 - 2) Trace-back the previous states:

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

a_{jk} : transition probability from previous state π_j to current state π_k

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

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

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

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 $X = \text{Shirt, Hoodie}$

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
- $x_1 = \text{Shirt}$
- 3 states: *Rainy, Cloudy, Sunny*

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

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

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

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 $t = 2$, $x_2 = \text{Hoodie}$

• 1) For Rainy

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

$$\max(V_{\text{Rainy}}(1)a_{\text{RainyRainy}}, V_{\text{Cloudy}}(1)a_{\text{CloudyRainy}}, V_{\text{Sunny}}(1)a_{\text{SunnyRainy}})$$

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

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

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 $t = 2$, $x_2 = \text{Hoodie}$

• 1) For Rainy

$$\begin{aligned}
 V_{\text{Rainy}}(2) &= e_{\text{Rainy}}(\text{Hoodie}) * \max_{j \in \{\text{Rainy}, \text{Cloudy}, \text{Sunny}\}} (V_j(1) a_{jk}) \\
 &= 0.01 * \max(0.48 * 0.6, 0.15 * 0.4, 0.001 * 0.1) = 0.01 * 0.48 * 0.6 = 0.0029
 \end{aligned}$$

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

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 $t = 2$, $x_2 = \text{Hoodie}$

• 2) For Cloudy

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

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

$$= 0.1 * (0.48 * 0.3, 0.15 * 0.3, 0.001 * 0.4) = 0.1 * 0.48 * 0.3 = 0.0144$$

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

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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 $t = 2$, $x_2 = \text{Hoodie}$

• 3) For Sunny

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

$$\max(V_{\text{Rainy}}(1)a_{\text{RainySunny}}, V_{\text{Cloudy}}(1)a_{\text{CloudySunny}}, V_{\text{Sunny}}(1)a_{\text{SunnySunny}})$$

$$= 0.79 * (0.48 * 0.1, 0.15 * 0.3, 0.001 * 0.5) = 0.79 * 0.48 * 0.1 = 0.0379$$

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

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

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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:

$$\begin{aligned} \operatorname{argmax} (V_{\text{Rainy}}(2), V_{\text{Cloudy}}(2), V_{\text{Sunny}}(2)) &= \operatorname{argmax} (V_{\text{Rainy}}(2), V_{\text{Cloudy}}(2), V_{\text{Sunny}}(2)) = \\ &= \operatorname{argmax}(0.0029, 0.0144, \mathbf{0.0379}) = 3, \text{ i.e. Sunny} \end{aligned}$$

Backtracking through the pointers:

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

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:

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

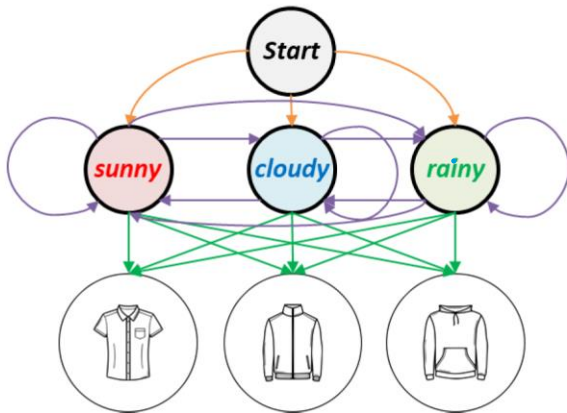
$$\pi_{i-1} = \operatorname{Ptr}_{\pi_i}(i)$$



HMM problem 3 (Learning)

HMM problem 3 – weather example

- HMM problem 3 (also called Learning problem):
- Given an observation sequence $X = x_1, x_2, \dots, x_M$, what is the model $\lambda = (\pi, A, A_0)$?
- Given the observation sequence $X = \text{Shirt}, \text{Hoodie}$ (a sequence of clothing you observed in the past days), **what is the model $\lambda = (\pi, A, A_0)$?**



Transition probability matrix A

| | | π_t | | |
|-------------|--------|---------|--------|-------|
| | | Rainy | Cloudy | Sunny |
| π_{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!

- Uses the Expectation Maximization (EM) algorithm – outline:
 1. 1. Initialize $\lambda = (\pi, A, A_0)$ to random values
 2. 2. Compute the probabilities A , and A_0 for all time steps, for all states π (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. 4. If $P(X | \lambda)$ have increased, return to step 2, otherwise stop.

- 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”)

- 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

- 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

- 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