Hidden Markov model (NLP)

Prof Nirali Arora

What is HMM

A Hidden Markov Model (HMM) is a statistical model used to represent systems that are assumed to follow a Markov process with hidden states.

- **Markov Process:** In a Markov process, the future state depends only on the present state and not on the sequence of events that preceded it. This property, known as the "Markov property," simplifies complex probabilistic calculations.
- **Hidden States:** In an HMM, the system being modeled has states that are not directly observable (hidden), but each state generates observable outputs. These outputs give us clues about the underlying hidden state.
- **Application Context:** HMMs are particularly powerful for modeling sequences of data, where the goal is often to infer the hidden states from the observable data. This makes HMMs suitable for a variety of tasks, especially in Natural Language Processing (NLP).

Statistical Foundation and Relevance in NLP

- **Probabilistic Framework: HMMs are grounded in probability theory, making them well-suited for dealing with uncertainty in data. They calculate the probability of sequences of observations and the likelihood of transitions between hidden states.**

Sequential Nature of Language: Language is inherently sequential, with the meaning of a word often depending on its context. HMMs can model these dependencies effectively, making them useful for tasks like Part-of-Speech (POS) tagging, speech recognition, and more.

Interpretability: HMMs provide a clear and interpretable framework where one can explicitly define the relationships between states and observations, making them easier to understand compared to more complex models like deep neural networks.

Components of HMM:

1. **States:**
   - **Hidden States (Q):** These are the states that the model transitions through, but they are not directly observable. In NLP, these could be the underlying grammatical roles (e.g., noun, verb) in a sentence.
   - **Observable States (O):** These are the outputs or observations that we can see. For example, in POS tagging, the words in a sentence are the observable states.
2. **Transition Probabilities (A):**
   - These probabilities define the likelihood of moving from one hidden state to another. The probability of transitioning from state iii to state jjj is denoted as AijA_{ij}Aij.
   - **Example:** In language, if the current state is "noun," the transition probability might tell us how likely it is to move to a "verb" in the next word.

Components of HMM

1. **Emission Probabilities (B):**
    - Emission probabilities describe the likelihood of an observable output given a particular hidden state. If the hidden state is known, these probabilities tell us the chance of observing a specific word or symbol.
    - **Example:** If the hidden state is "noun," the emission probability might tell us how likely it is that the observed word is "dog."
2. **Initial State Distribution (π):**
    - This represents the probability distribution over the possible initial hidden states at the beginning of the sequence.
    - **Example:** At the start of a sentence, the initial state distribution might tell us the likelihood of the sentence starting with a noun, verb, etc.

Example on HMM

Rahul has three main activities he might do each day:

1. **Jogging**
2. **Going to the office**
3. **Cleaning his residence**

What Rahul decides to do on any given day depends on the weather. However, the weather conditions are not directly observable by Ashok. Ashok only knows what activity Rahul did, but he wants to infer what the weather was like based on these activities.

Hidden States (Weather Conditions):

- The weather conditions are the **hidden states** because Ashok cannot directly observe the weather. Let's assume the possible weather conditions are:
    - **Sunny**
    - **Rainy**
    - **Cloudy**

Observable States (Rahul’s Activities):

- Rahul’s activities are the **observable states** that Ashok can see. These activities are:
  - **Jogging**
  - **Going to the office**
  - **Cleaning the residence**

Transition Probabilities (Weather Changes):

- The **transition probabilities** represent how likely the weather is to change from one day to the next. For example:
  - If today is sunny, there might be a high probability that tomorrow will also be sunny.
  - If today is rainy, there might be a higher probability that tomorrow will be cloudy or stay rainy.
- These probabilities tell us the likelihood of moving from one hidden state (weather condition) to another.

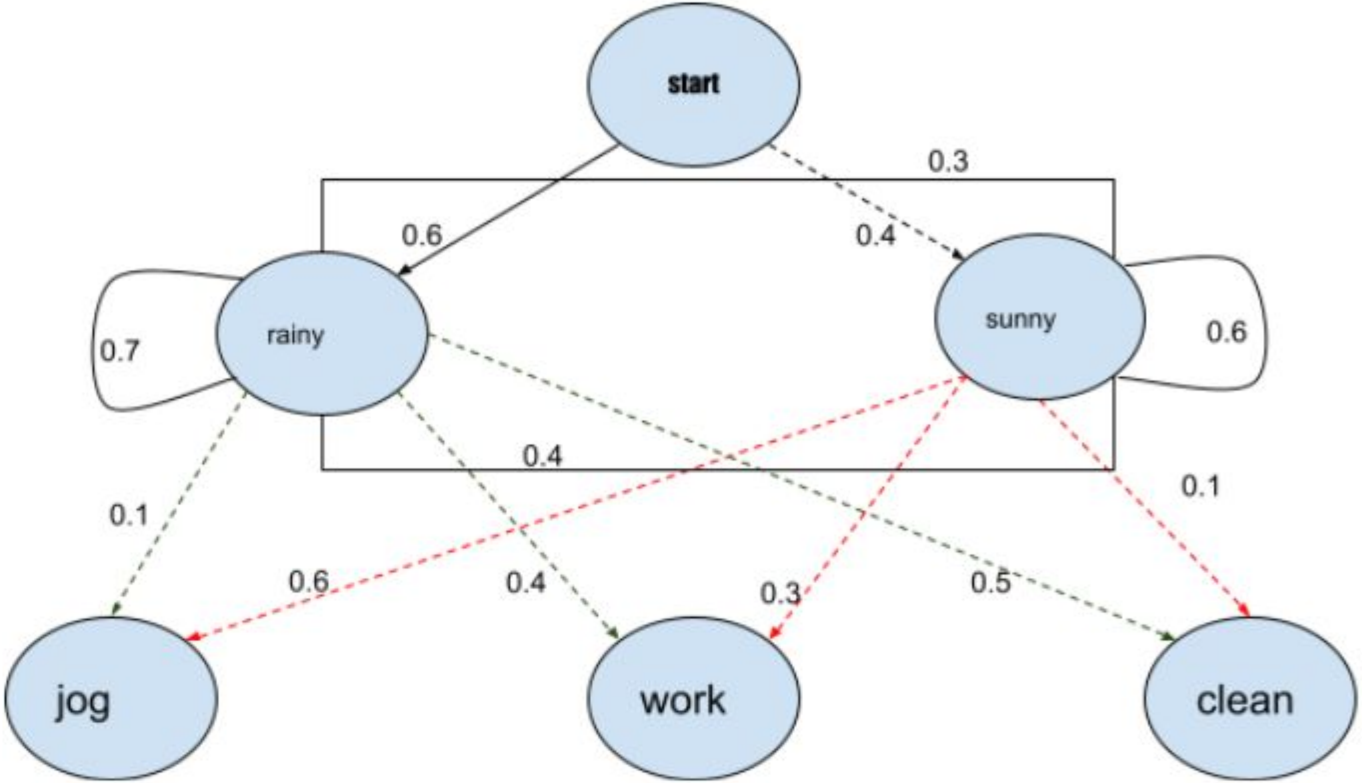
Emission Probabilities (Activity Given Weather):

- The **emission probabilities** represent how likely Rahul is to perform each activity given a specific weather condition. For example:
  - On a **sunny** day, Rahul might be more likely to go jogging.
  - On a **rainy** day, he might be more likely to stay indoors and clean his residence.
  - On a **cloudy** day, he might go to the office.

Initial State Distribution:

- This is the probability distribution over the possible weather conditions on the first day. For example, on day one, there could be a 50% chance that the weather is sunny, a 30% chance it’s cloudy, and a 20% chance it’s rainy.

HMM


start
0.3
0.6
0.4
rainy
sunny
0.7
0.6
0.4
0.1
0.1
0.6
0.4
0.3
0.5
jog
work
clean

How to use HMM for pos tagging

Suppose we have a simple sentence: **"The cat sleeps."**

Our goal is to determine the correct part of speech for each word in the sentence using an HMM.

Define the Components of the HMM

1. **Hidden States (Parts of Speech):**
   - These are the possible parts of speech that the words in the sentence could belong to. In this case, the hidden states might include:
     - **Noun (N)**
     - **Verb (V)**
     - **Determiner (D)**

Observable States (Words):

- These are the actual words in the sentence that we can see:
  - "The"
  - "cat"
  - "sleeps"

These probabilities represent the likelihood of transitioning from one part of speech to another. For example:

- P(Noun→Verb)
- P(Determiner→Noun)
- P(Determiner→Noun)

These probabilities are learned from a tagged corpus, where similar sentences have been labeled with their corresponding parts of speech

Emission Probabilities (B):

- These probabilities represent the likelihood of a word being a particular part of speech. For example:
  - $P(\text{"cat"} \mid \text{Noun})$
  - $P(\text{"sleeps"} \mid \text{Verb})$
  - These probabilities are also learned from a tagged corpus.

Initial Distribution

- This represents the probability of each part of speech being the first in a sentence. For example:
  - P(D)=0.6 (There's a 60% chance that a sentence starts with a determiner)
  - P(N)=0.4 (There's a 40% chance that a sentence starts with a noun)
  -

Apply the HMM to the Sentence

Sequence of Words:

- "The cat sleeps"

Objective:

- Determine the most likely sequence of parts of speech (POS tags) for the sentence.

Calculate Using the Viterbi Algorithm

The Viterbi algorithm is used to find the most likely sequence of hidden states (POS tags) given the observed words.

1. **Initialization:**
   - For the first word "The," calculate the probability for each possible part of speech:
     - $P(D) \times P(\text{"The"} \mid D)$
     - $P(N) \times P(\text{"The"} \mid N)$
     - $P(V) \times P(\text{"The"} \mid V)$

Recursion:

- For the second word "cat," calculate the probability for each POS ta
  considering all possible previous states (POS tags for "The"):
    - $P(\text{"cat"}|N) \times [P(D) \times P(\text{"The"}|D) \times P(D \rightarrow N)]$
    - Similar calculations for other transitions and emissions.

Termination:

- For the final word "sleeps," calculate the probability for each possible POS tag, considering all possible previous states:
  - $P(\text{"sleeps"}|V) \times (\text{max probability path from previous steps})$

Traceback:

- Backtrack through the most likely transitions to determine the most likely sequence of POS tags.

Step 4: Result

After applying the Viterbi algorithm, you might get the following most likely sequence of POS tags for the sentence "The cat sleeps":

- "The" → Determiner (D)
- "cat" → Noun (N)
- "sleeps" → Verb (V)

Summary

Hidden States: Parts of speech (D, N, V)

Observed States: Words in the sentence ("The," "cat," "sleeps")

Transition Probabilities: Likelihood of moving from one POS to another

Emission Probabilities: Likelihood of a word given a POS

Viterbi Algorithm: Used to find the most likely sequence of POS tags.