HIDDEN MARKOV MODEL

By Neha Gupta

A Hidden Markov Model (HMM) is a statistical model which is also used in machine learning. It can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable.

A simple example of an HMM is predicting the weather (hidden variable) based on the type of clothes that someone wears (observed).

By forecasting weather precisely we can prevent and overcome many hazards that could lead to a great loss to a nation. HMM is used to predict weather using the Markov Chain property. The training of the model and probability of occurrence of an event is calculated by observing weather data for the last 21 years. The data is firstly categorized based on standard values set apart. The result obtained shows that our model is reliable and works very well in predicting the weather for the next 5 days based on today's weather pattern.

What is Markov Model?

The Markov process assumption is simply that the “future is independent of the past given the present”. In other words, assuming we know our present state, we do not need any other historical information to predict the future state.

Important terms and definitions:

1. **Transition data** — the probability of transitioning to a new state conditioned on a present state.

2. **Emission data** — the probability of transitioning to an observed state conditioned on a hidden state.

3. **Initial state information** — the initial probability of transitioning to a hidden state. This can also be looked at as the prior probability.

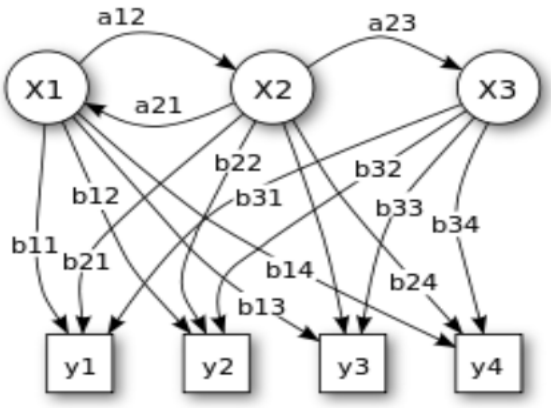


Figure 1. Probabilistic parameters of a hidden Markov model (example)
X — states
y — possible observations
a — state transition probabilities
b — output probabilities

Need for use:

1. Modeling sequences of data.
2. Used in stock prices, credit scoring, and webpage visits.
3. In Computational Genomic Annotation. It includes structural annotation for genes and other functional elements and functional annotations for assigning functions to the predicted functional elements.

Whether HMMs are supervised or unsupervised?

- Unsupervised: When we want to discover a pattern or model a distribution but don't have any labeled data.
- Supervised: When we do have labels and want to be able to accurately predict the labels.
- HMMs are used for modelling sequences:
    - x(1), x(2), x(3), x(4).…….x(t),....,x(T).
    - No labels are available.

Classification (an example of HMM):

- How can we classify the voices into male and female categories using HMM?
- The key Idea is Bayes' rule.
- We model P(x | male) and P(x | female)
- P(x | male): Collect all-male data and train an HMM
- P(x | female): Collect all-female data and train another HMM.

Applications of Hidden Markov Model:

Hidden Markov models are known for their applications to thermodynamics, statistical mechanics, physics, chemistry, economics, finance, signal processing, information theory, pattern recognition - such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges

Applications, where the HMM can be used, have sequential data like time series data, audio, and video data, and text data or NLP data.

Conclusion over its Nature:

- Hidden Markov Models are **Unsupervised** in Nature.

- It is used for Classification using Bayes' rule and creating a separating model for each class, and choosing the class that gives us the maximum posterior probability.

Markov Property:

- When tomorrow's weather depends on today's but not on yesterday's weather
- When the next word in the sentence depends only on the previous word in the sentence but not on any other words.
- In general, our assumption is that the current state depends only upon the previous state or the next state depends only upon the current state.

Another way of saying is:
The distribution of state at a time 't' depends only on the state at a time 't-1'.

- In general: 'states'
- State at time t: s(t)
- p(s(t) | s(t-1), s(t-2),.....,s(0)) = p(s(t) | s(t-1))

We want to model the joint probability i.e. probability of a sequence of states, which becomes (using chain rule of probability):

p(s4, s3, s2, s1) = p(s4 | s3, s2, s1) * p(s3, s2, s1)
= p(s4 | s3, s2, s1) * p(s3 | s2, s1) * p(s2, s1)
= p(s4 | s3, s2, s1) * p(s3 | s2, s1) * p(s2 | s1) * p(s1)
= p(s4 | s3) * p(s3 | s2) * p(s2 | s1) * p(s1)
(assuming Markov property)

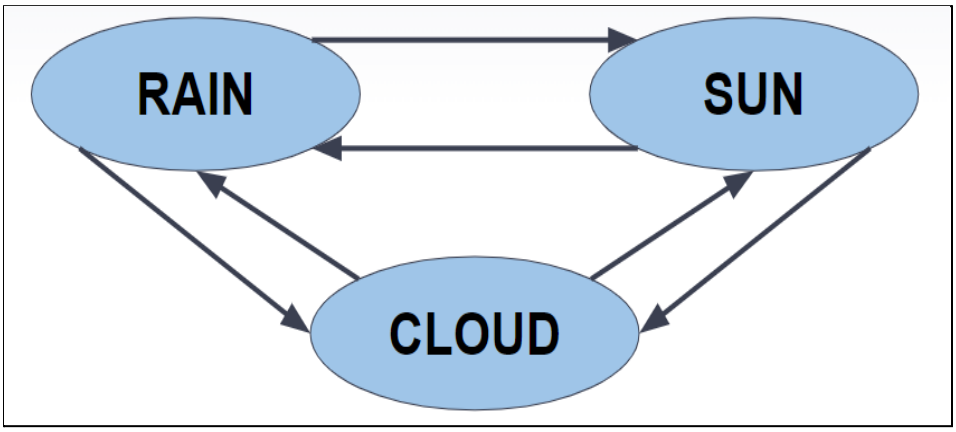
Markov Models are often used to model the probabilities of different states and the rates

of transition among them.

- First Order Markov: p(s(t) | s(t-1))
- Second Order Markov: p(s(t) | s(t-1), s(t-2))

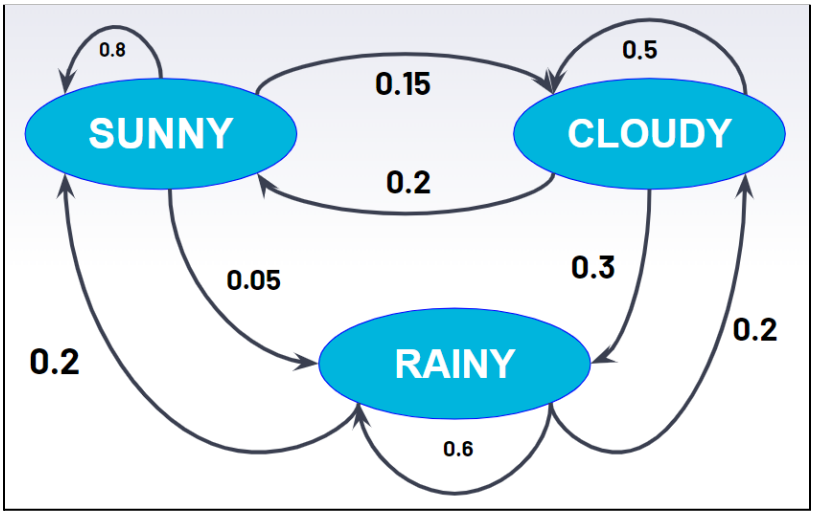
MARKOV MODEL EXAMPLE:

Weather Example:
3 states: SUN, RAIN, CLOUD



- Here, each node is a state and each edge is the probability of going from one state to the next state.

- This is 1st order Markov Model because weights depend only on the current state and it only affects the next state.



HOW MANY WEIGHTS DO WE NEED TO CALCULATE?

Each state can go to each state, including itself.

M States —-----> M* M weights
A —-----> M X M Matrix
A(i, j) = p (s(t) = j | s(t-1) = i)
Constraints: A(i,:) must sum to 1; i = 1,....., M
Starting Position: Where do you start?
—--> we need to model p(s(0))
We usually store it in the symbol pi
1 X M row vector

EXAMPLE:

Q) What is the probability of the sequence? (sun, sun, rain, cloud)

p(sun, sun, rain, cloud) = p(cloud | rain) * p(rain | sun) * p(sun | sun) * p(sun)
= 0.2 * 0.05 * 0.8 * p(sun)

In general:
p(s(0),.....,s(T)) = π(s(0))

How is a Markov Model trained?

This can be done by using **maximum likelihood.**

For example: Suppose we have training data of 3 sentences.
1) I like dogs
2) I like cats
3) I love kangaroos

6 states: 0 = I, 1 = like, 2 = love, 3 = dogs, 4= cats, 5 = kangaroos

If we use maximum likelihood, then our initial state distribution is just one hundred percent probability.

π = [1, 0, 0, 0, 0, 0]

p(like | I) = ⅔
p(love | I) = ⅓
p(dogs | like) = p(cats | like) = ½
p(kangaroos | love) = 1

Markov Chain:

A Markov chain is a mathematical system that experiences transitions from one state to another according to certain probabilistic rules. The defining characteristic of a Markov

chain is that no matter how the process arrived at its present state, the possible future states are fixed.

Markov Chain: Example

Example: What is the probability of a sunny day 5 days from now?

P(sun(1)) = P(sun(1),sun(0)) + P(sun(1),rain(0)) + P(sun(1),cloud(0))

= P(sun(1)|sun(0)) Π (sun)
+
P(sun(1)|rain(0)) Π (rain)
+
P(sun(1)|cloud(0)) Π (cloud)

In General:
$P(s(1)) = \Pi A$
$P(s(2)) = \Pi A A = \Pi A^2$
$P(s(t)) = \Pi A^t$

Stationary Distribution

What if P(S) = P(S) A?

I.e. We end up with the same state distribution.

The state distribution never changes-> Stationery

=> This is just the eigenvalue problem where the eigenvalue is 1.

Issues associated with it:

- Eigenvalues are not unique -> make it sum to 1 so it's a proper distribution.
- P(s) is a row vector, typically a solver will solve $Av = \lambda v$
  (so transpose it first)

Limiting Distribution:

It is the state distribution that you settle into after a very long time.

What is the final state distribution?
$\Pi_\infty = \Pi A^\infty$
(Called a limiting distribution or equilibrium distribution)

But $A^\infty A = A^\infty$

So, $\Pi_\infty = \Pi_\infty A$

This means Π is also a stationary distribution.

Conclusion:

- If I take an Equilibrium distribution multiplied by A and get the same distribution, then it's a stationary distribution

- So, All the Equilibrium distributions are stationary distributions, but vice-versa is not true.

FROM MARKOV MODEL TO HIDDEN MARKOV MODEL

- Unsupervised ML (Cluster Analysis) and Unsupervised DL (hidden/latent variables)
- K-Means Clustering, Gaussian Mixture models, principal components analysis.
- Observations are stochastic
- Hidden causes are a stochastic tool

HMM, Real-life Example:

Suppose you are at a carnival, magician has 2 coins hidden behind his back.
He will choose one coin to flip at random, you can only see the result of the coin flip (H/T)
H/T - space of observed values
Hidden State - Which coin it is.
(This is a stochastic/random process)

The intuition behind HMMs:

HMMs are probabilistic models. They allow us to compute the joint probability of a set of hidden states given a set of observed states. The hidden states are also referred to as latent states. Once we know the joint probability of a sequence of hidden states, we determine the best possible sequence i.e. the sequence with the highest probability, and choose that sequence as the best sequence of hidden states.

HMM:

It has 3 parts: $\mathbf{\Pi}$, A, B
Π_i: the probability of starting at state i
A(i, j) = probability of going to state j from state i.
B(j, k) = probability of observing symbol k in state j.

Example 1: Magician really likes coin 1, $\pi_1 = 0.9$
Example 2: Magician is fidgety, A(2,1) = 0.9
A(1, 2) = 0.9
Here, A is a **state transition matrix**
In HMM, the state themselves are hidden.

HMM for Stock Price Prediction:

The hidden Markov model has been widely used in the financial mathematics area to predict economic regimes or predict stock prices.

Choosing a number of hidden states for the HMM is a critical task. We use four common criteria: the AIC, the BIC, the HQC, and the CAIC to evaluate the performances of HMM with different numbers of states. These criteria are suitable for HMM because, in the model training algorithm, the Baum–Welch Algorithm, the EM method was used to maximize the log-likelihood of the model. We limit the number of states from two to six to keep the model simple and feasible for stock prediction.

AIC = -2 ln(L) + 2k
BIC = -2 ln(L) + k ln(M)
HQC = -2 ln(L) + k ln(ln(M))
CAIC = -2 ln(L) + k(ln(M) + 1)
where L is the likelihood function for the model, M is the number of observation points, and k is the number of estimated parameters in the model.We assume that the distribution corresponding with each hidden state is a Gaussian distribution, therefore, the number of parameters, k, is formulated as k = N2 + 2N - 1, where N is numbers of states used in the HMM.

Independence Assumptions:

- More than just Markov's assumption
  The next state depends only on the previous state

- Observed ‘k’ depends only on state j.
  Does not depend on the state at any other time
  Does not depend on any other observation

Q) How to choose the number of hidden states:

- It is a hyperparameter
- It can be done by Cross-Validation

N training samples + N parameters —--> can achieve the perfect score

But this doesn't say anything about how well the model will perform on the unseen dataset.

How to avoid overfit and underfit?

- Leave some data out of training (validation set)
- Compute the Cost of the Validation set
- Choose the number of hidden states that give the highest validation accuracy

---