**Markov Models:**

Prior to the discussion on Hidden Markov Models, it is necessary to consider the broader concept of a Markov Model. A Markov Model is a stochastic state space model involving random transitions between states where the probability of the jump is only dependent upon the current state, rather than any of the previous states. The model is said to possess the Markov Property and is "memoryless". Random Walk models are another familiar example of a Markov Model.

The simplest model, the Markov Chain, is both autonomous and fully observable. It cannot be modified by actions of an "agent" as in the controlled processes and all information is available from the model at any state. A good example of a Markov Chain is the Markov Chain Monte Carlo (MCMC) algorithm used heavily in computational Bayesian inference.

Once the system is allowed to be "controlled" by an agent(s) then such processes come under the heading of Reinforcement Learning (RL), often considered to be the third "pillar" of machine learning along with Supervised Learning and Unsupervised Learning. If the system is fully observable, but controlled, then the model is called a Markov Decision Process (MDP). A related technique is known as Q-Learning[11], which is used to optimise the action-selection policy for an agent under a Markov Decision Process model. In 2015 Google DeepMind pioneered the use of Deep Reinforcement Networks, or Deep Q Networks, to create an optimal agent for playing Atari 2600 video games solely from the screen buffer.

**Markov Model Mathematical Specification:**

In quantitative finance the analysis of a time series is often of primary interest. Such a time series generally consists of a sequence of T discrete observations X1, …... , XT . An important assumption about Markov Chain models is that at any time t, the observation XT captures all of the necessary information required to make predictions about future states. This assumption will be utilised in the following specification.

Formulating the Markov Chain into a probabilistic framework allows the joint density function for the probability of seeing the observations to be written as:

This states that the probability of seeing sequences of observations is given by the probability of the initial observation multiplied T-1 times by the conditional probability of seeing the subsequent observation, given the previous observation has occurred. It will be assumed in this article that the latter term, known as the transition function, p(XT | XT-1) will itself be time-independent.

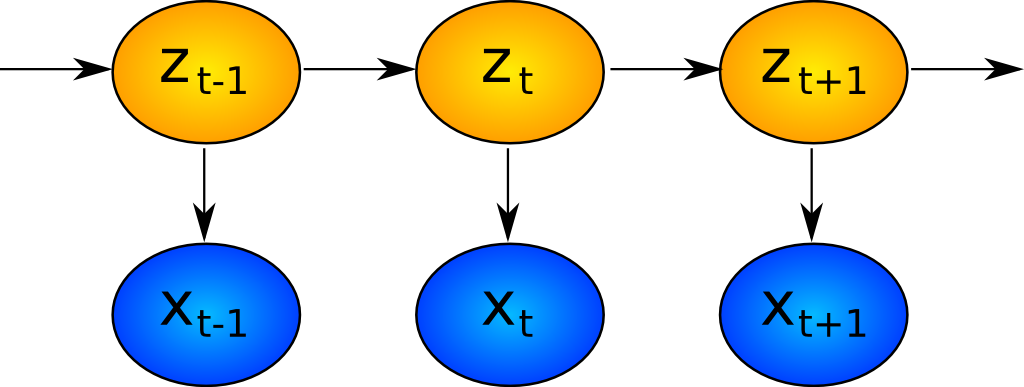
**Hidden Markov Model Mathematical Specification:**

The corresponding joint density function for the HMM is given by (again using notation from Murphy (2012)

In the first line this states that the joint probability of seeing the full set of hidden states and observations is equal to the probability of simply seeing the hidden states multiplied by the probability of seeing the observations, conditional on the states. This makes sense as the observations cannot affect the states, but the hidden states do indirectly affect the observations.

The second line splits these two distributions into transition functions. The transition function for the states is given by p(zt | zt-1) while that for the observations (which depend upon the states) is given by p(xT|zT).

In order to make this a little clearer the following diagram shows the evolution of the states and how they lead indirectly to the evolution of the observations, xt :



**Filtering of Hidden Markov Models:**

With the joint density function specified it remains to consider the how the model will be utilised. In general state-space modelling there are often three main tasks of interest: Filtering, Smoothing and Prediction.

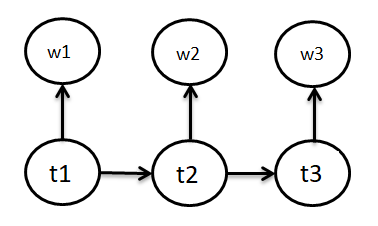
* Prediction - Forecasting subsequent values of the state
* Filtering - Estimating the current values of the state from past and current observations
* Smoothing - Estimating the past values of the state given the observations

Filtering and smoothing are similar, but not identical. Smoothing is concerned with wanting to understand what has happened to states in the past given current knowledge, whereas filtering is concerned with what is happening with the state right now.

**NLP: Hidden Markov Model for POS Tagging**

Part of Speech (POS) Tagging is an important NLP application. The objective in such problems is to tag each word in a given sentence with an appropriate POS (noun, verb, adverb, etc.). In the model, we have tag transition probabilities, i.e. given that the tag of the previous word is said ti-1, the ti will be the tag of the current word. And the second concept of the model is the probability that given the tag of the current word is ti, the word will be wi. To state more clearly: Hidden Markov Model is a type of Generative (Joint) Models in which the hidden states (here POS tags) are considered to be given and observed data is considered to be generated.

The ti in the figure below refers to the POS tags of the state and wi refers to the words emitted by the states.



Now, let’s see the element of the model minutely to see the Markov process in it even more clearly. The elements of the Hidden Markov Model are:

A set of states: here POS Tags

The output from each state: here ‘word’

Initial state: here beginning of the sentence

State transition probability: here P(tn | tn-1)

Apart from being particularly useful in this aspect of NLP, Finite State Transducers and multiple other algorithms are based on Markov Chains.

**References:**

* <https://www.quantstart.com/articles/hidden-markov-models-an-introduction/>
* [1] (2016). Markov Model, Wikipedia
* [2] (2016). Hidden Markov Model, Wikipedia
* [5] (2016). Markov Chain, Wikipedia
* https://www.analyticsvidhya.com/blog/2021/02/markov-chain-mathematical-formulation-intuitive-explanation-applications/#:~:text=NLP:%20Hidden%20Markov%20Model%20for%20POS%20Tagging%20Part,with%20an%20appropriate%20POS%20(noun,%20verb,%20adverb,%20etc.).