**Markov Models**

Created: 10/04/2023 by Tom Lever

Updated: 10/04/2023 by Tom Lever

*Machine Learning*

Designing algorithms for inferring what is unknown based on knowns

Blend of Statistics and AI

Used in recognizing spam, recognizing handwriting, self-driving cars, blurring faces in Google Street View imagery, recognizing speech, recommending, and interpolation of climate data

Supervised vs. Unsupervised Learning

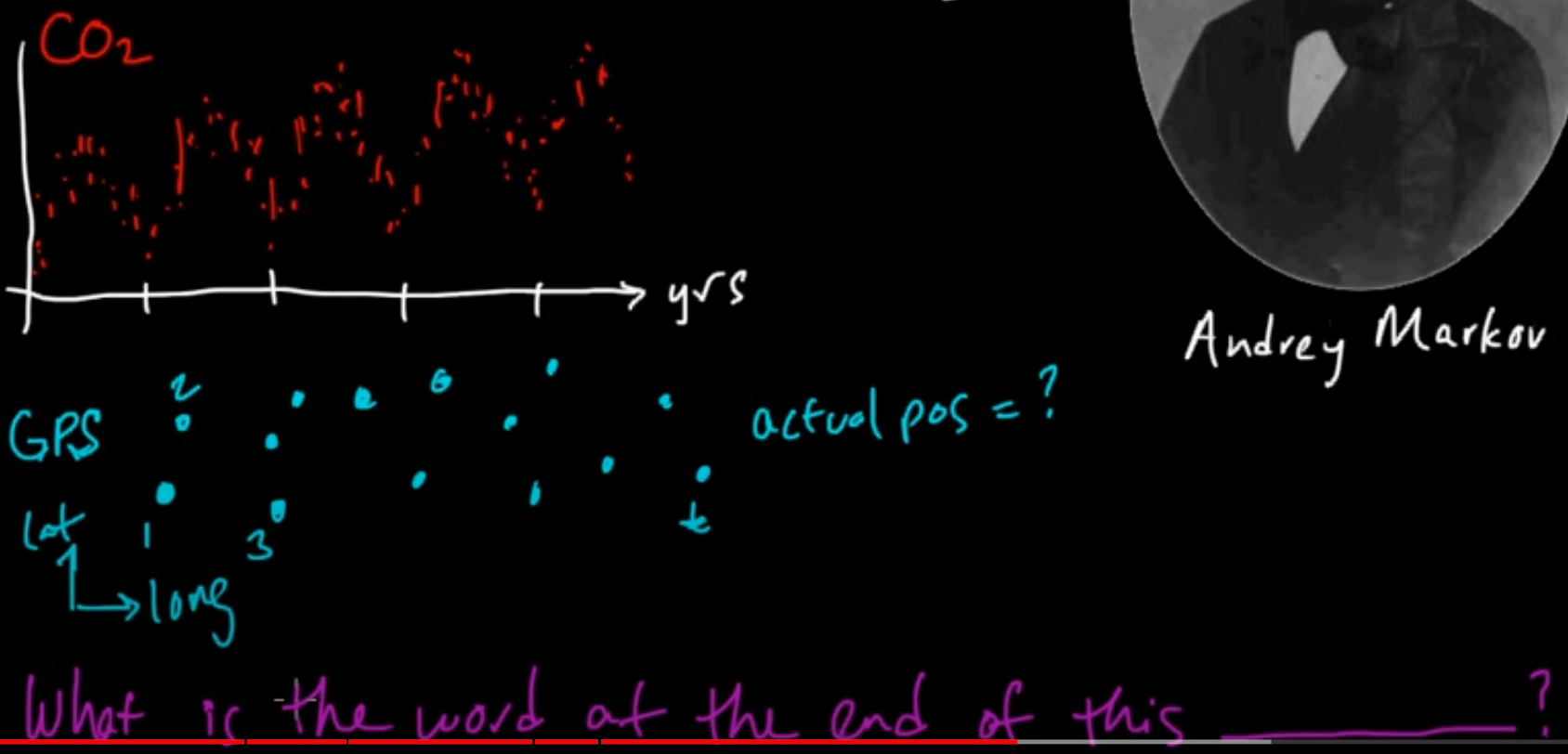
Supervised learning is a problem in which we choose a function based on a sequence of coordinate pairs where is a data point and is a target value that can be used to approximate a target value based on a data point .

Types of supervised learning include classification and regression. For classification, belongs to some finite set. For regression, .

The future is independent of the past, given the present.

Markov models deal with temporal data related to subjects like weather, finance, language, music.

Consider carbon-dioxide concentration in atmosphere vs. year, position over time, and completing sentences.



Let be a number of data points (e.g., ). Consider tuple of sequential data . We model this data using random variables in . These random variables are not independent and identically distributed: carbon-dioxide concentration a little after a time is close to the carbon dioxide concentration at that time. The most accurate prediction of what will happen in the near future is what is happening now; the most accurate prediction of what is happening now is what happened in the recent past. Suppose depends on where are instants in time separated by a certain unit of time.

We assume that data points in occur at discrete instants in time. We assume that the possible values of the data points in comprise a finite set. Discrete random variables in form a first-order discrete-time Markov chain if the joint distribution of the random variables respect



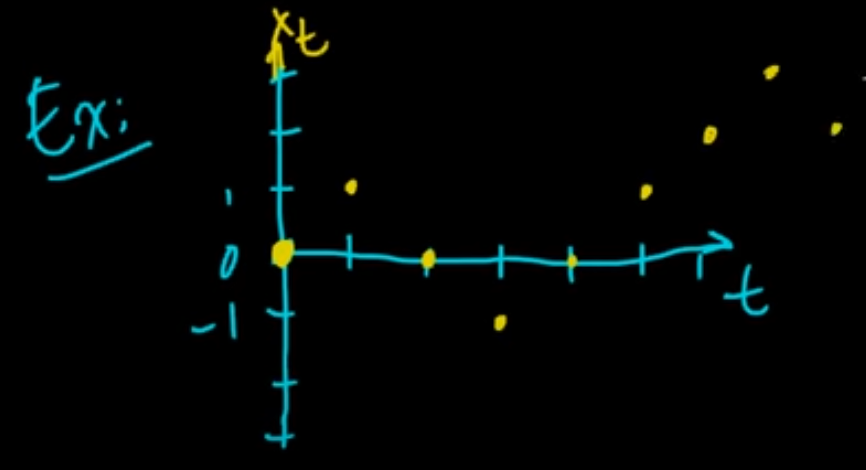
or the joint distribution factors as

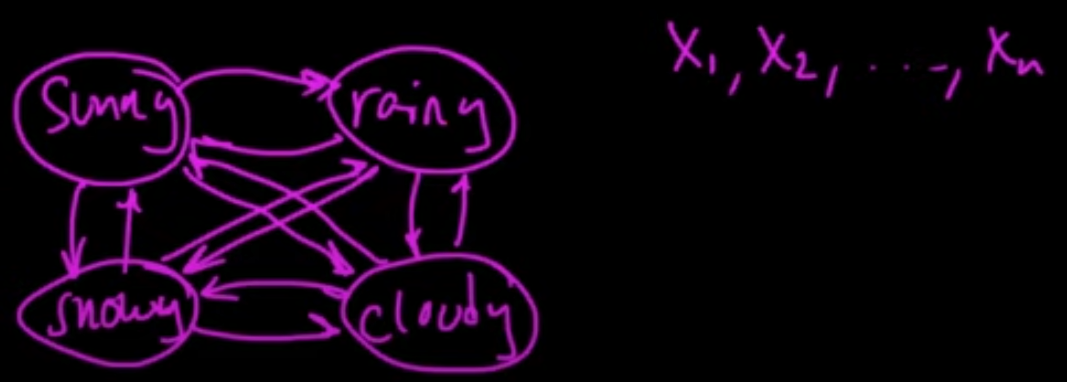
or

or is conditionally independent of given .

For example, is conditionally independent of given .

First-order discrete-time, discrete-space Markov chain (e.g., random walk on integers, weather)

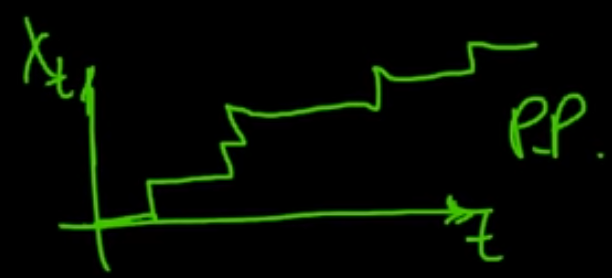




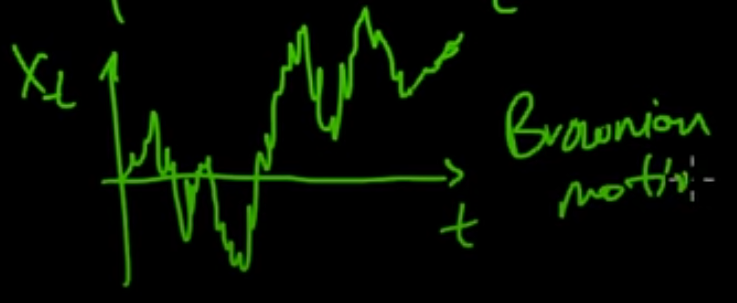
Second-order discrete-time, discrete-space Markov chain



First-order continuous-time, discrete-space Markov process (e.g., a Poisson process)



First-order discrete-time, continuous-space / state-space Markov process (e.g., 1D Brownian motion)

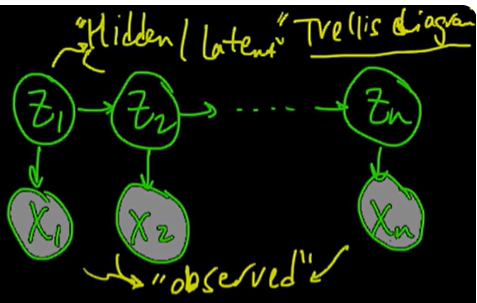


In our first graph above, the nodes represent the state of the system as it evolves. We assumed that our data is a subset of the true state of the system as it evolved. However, we can’t expect to perfectly observe the complete true state of the system. We collect noisy observations at instants of time that different from the true values of the system at those times. We can theorize that there are hidden / latent information / variables; the state of the system consists of an observed state and a hidden state. We can construct a Hidden Markov Model.

Consider a tuple of observed random variables . Each random variable takes on a discrete value, a real value, a vector of real values, etc. Data is a tuple of observed values.

Consider a tuple of hidden random variables . Each random variable takes on a discrete value in . Each value of a hidden variable occurs at a discrete time.

These variables / the joint distribution of these variables respects a trellis diagram / graphical model for a Hidden Markov Model:



Probability of transition from state to state .

Let be a number of states in a Hidden Markov Model (e.g., ).

Emission probability

Emission probability

is a probability density distribution of a set of real values, vector of real values, etc.

is a probability mass distribution of a set of discrete values

Initial probability mass distribution

If each take one of a set of discrete values, may be a probability mass distribution. If take one of a set of countably infinite values, may be a geometric or Poisson distribution. If takes a real value, may be a Gaussian distribution. The form of the distribution is arbitrary and is not as important as the fact that the joint distribution factors as above and respects the graphical model above to performing tractable inference using the Hidden Markov Model.

Suppose that , a finite set.

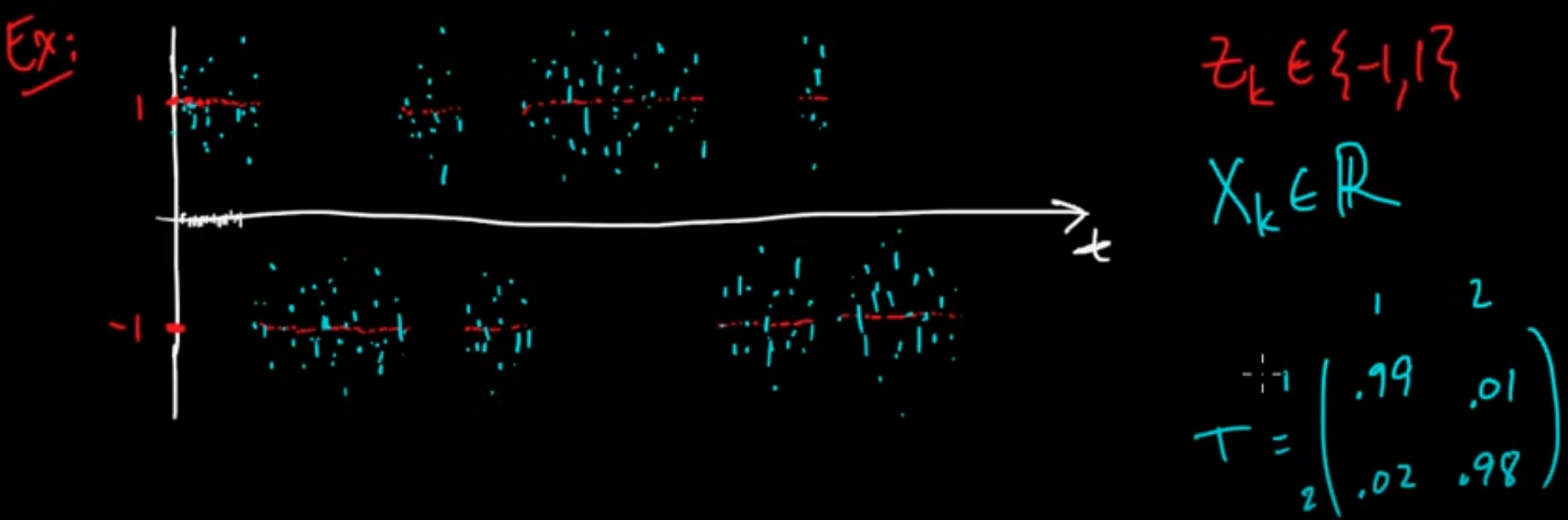
Suppose that .

Suppose that is normally distributed about .

Suppose that is conditionally independent of everything else given .

Red is .

Blue is .



Dynamic Programming (Programming means Optimization)

Efficiency of dynamic programming comes from reusing previous computational results.

Assume we know emission probabilities , transition probabilities , and initial probability .

Let .

Let .

The forward / backward algorithm computes .

The forward algorithm computes .

The backward algorithm computes .

The elements of are conditionally independent of given .

Change detection involves calculating .

We can estimate emission probabilities, transition probabilities, and initial probability using the Baum-Welch algorithm that combines the forward / backward algorithm and expectation maximization.

We can sample from the posterior probability distribution .

*Forward Algorithm for Hidden Markov Model*

The forward algorithm computes

is conditionally independent of given .

is conditionally independent of given .

To resolve underflow,

An algorithm is big- of some function ; i.e., , if there exist constants such that for all large greater than , time that the algorithm takes “on argument ” is less than or equal to upper bound , and grows proportionally with .

An algorithm if big- of some function ; i.e., , if there exist constants such that for all large greater than , time that the algorithm takes “on argument ” is less than or equal to upper bound , is greater than or equal to , and grows proportionally with .

Tight bound on time for each

Tight bound on time for each

Tight bound on time for the forward algorithm. Time that forward algorithm takes grows proportionally with .

The forward algorithm computes according to

and has tight bound on time .

The forward / backward algorithm computes

A naïve approach to computing involves computing

The time needed to compute is .

*Backward Algorithm*

We are given data . We assume that we know emission probabilities , transition probabilities , and initial probability mass distribution .

Our goal is to compute .

is conditionally independent of and given .

is conditionally independent of given .

Time complexity

*Viterbi Algorithm*

The forward / backward algorithm can be used to do inference on “the ’s”, the hidden states, given “a sequence of ’s”, the observed states, to .

The Viterbi algorithm is used to find the most probable sequence of hidden states (e.g., characters) for a given sequence of observed states (e.g., images of alphabetic letters).

We are given data . We assume that we know emission probabilities , transition probabilities , and initial probability mass distribution .

Most likely sequence .

If and , then .

*Rejection Sampling*

Way to generate independent samples from a probability density distribution / function that is potentially unnormalized.

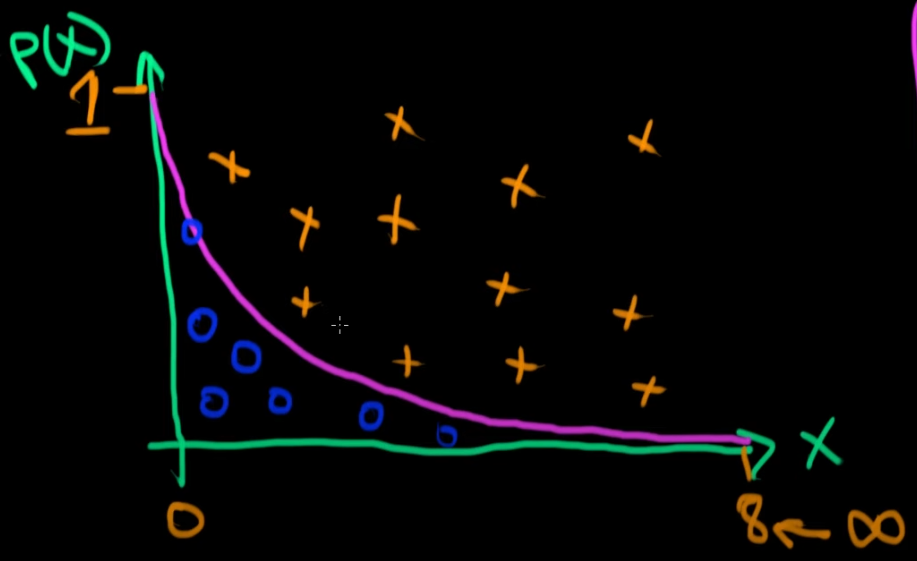
Suppose we are generating samples from exponential distribution with rate parameter .

Density .

Having an analytic density does not imply that I can independently sample from that density.

A PDF describes a histogram of samples, but not how to generate samples.

Rejection sampling is a way given a PDF to generate independent samples.



We generate a uniformly distributed between and or between and .

We generate some uniformly distributed between and .

If , we accept as a sample from our distribution. Otherwise, we reject .

By conducting rejection sampling, we collect a sample of accepted , .