INM431 Machine Learning

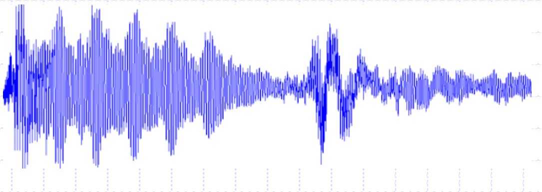
Artur S. d'Avila Garcez
a.garcez@city.ac.uk
http://www.staff.city.ac.uk/~aag/

Based on C. Bishop's book

Sequential Data

Introduction
Markov models
Hidden Markov models (HMMs)
- Definition
- Learning
- Inference
- Extensions

State space models / Linear dynamical systems



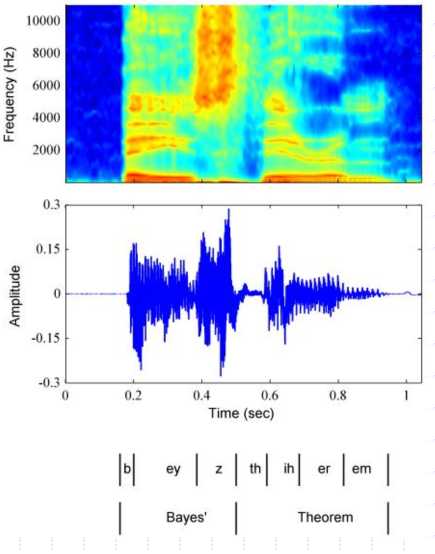
Intro – Sequential Data (1)

So far we have focused on sets of data points that were assumed to be **independent and identically distributed** (i.i.d.)

For many applications, the i.i.d. assumption is a bad one – such as when modelling **sequential data**

Sequential data often refer to **time series**, although they also cover other data types (e.g. DNA sequences, text)

Intro – Sequential Data (2)

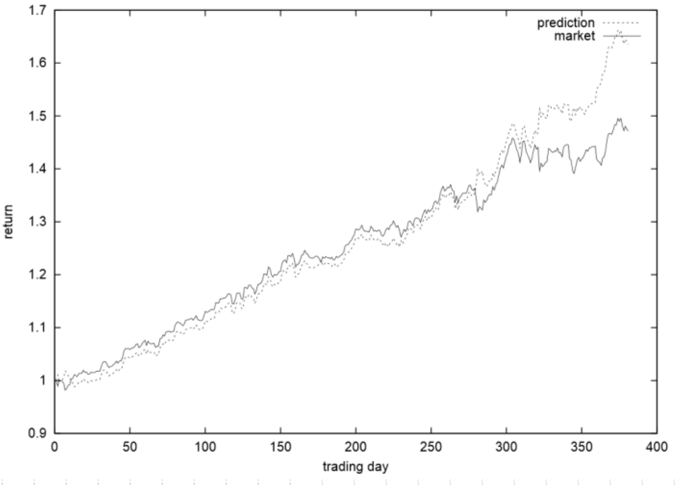


Application: speech recognition

Main assumption: successive observations are correlated

Intro – Sequential Data (3)

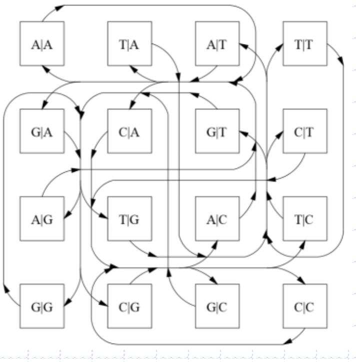
Application: prediction of financial time series



Intro – Sequential Data (4)

Application: modelling DNA sequences

...ApCpCpApTpGpApTpGpCpApGpGpApCpTpCpGpCpGp...
...| | | | | | | | | | | | | | | | | | | | ...
...TpGpGpTpApCpTpApCpGpTpCpCpTpGpApGpCpGpCp...



Intro – Sequential Data (5)

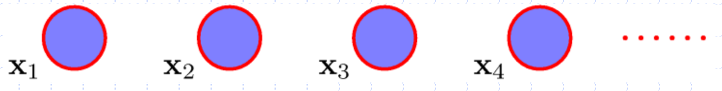
General assumptions:

- Modelling the next value in a sequence given observations of previous values
- Recent observations are likely to be more informative than historical observations
- Stationarity (model does not change/evolve over time)

But it's impractical to make a future observation depend on all previous observations (model would be too complex!)

Markov Models (1)

The simplest way to model a sequence of observations is to treat them as independent:



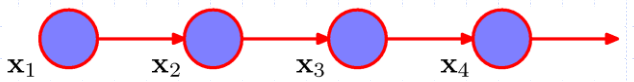
But this would fail to exploit the correlations between neighbouring observations – e.g. observing whether or not it rains today can help predicting if it will rain tomorrow.

We can relax the i.i.d. assumption by considering a Markov model:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$$

Markov Models (2)

If we assume that each current observation only depends on the most recent ("Markov assumption"), we obtain a **first-order Markov chain**:

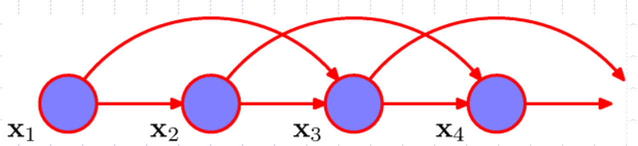


$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

State transition diagram for a 2-state Markov chain:

Markov Models (3)

We can also define a **second-order Markov chain**:



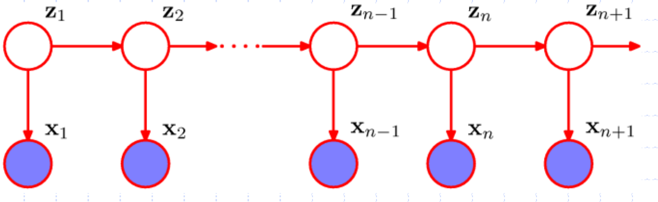
We can similarly consider extensions to an **M-th order Markov chain**...

...but there is a computational price for this increased flexibility. If we assume K states, then such a model would have $K^{M-1}(K-1)$ parameters.

Hidden Markov Models (1)

We can however create a model for sequences not limited by the Markov assumption, using only a limited number of parameters.

This can be achieved by introducing **latent variables** – linking each observation with a hidden state (which might be of a different type or dimensionality than the observation).

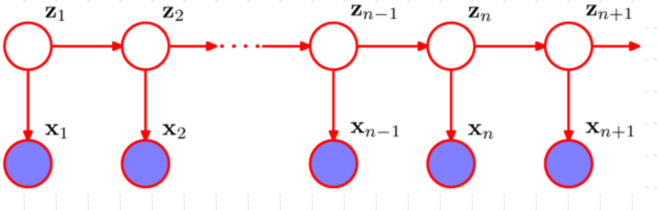


Hidden Markov Models (2)

The joint distribution for this model is given by:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = p(\mathbf{z}_1) \left[\prod_{n=2}^{N} p(\mathbf{z}_n | \mathbf{z}_{n-1}) \right] \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{z}_n)$$

- If the latent variables are discrete, we obtain a **Hidden Markov Model (HMM)**
- If the latent variables are continuous, we obtain a **State Space Model (SSM)**



Hidden Markov Models (3)

Elements of a HMM:

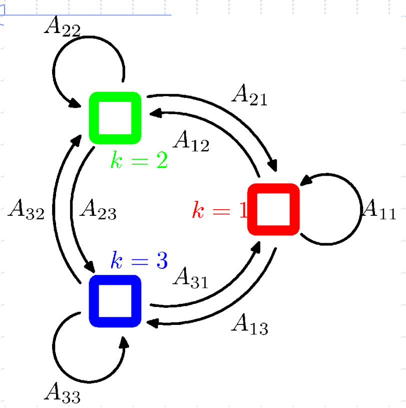
1. Transition probabilities A: $A_{jk} \equiv p(z_{nk} = 1 | z_{n-1,j} = 1)$
2. Prior probabilities: $\pi_k \equiv p(z_{1k} = 1)$
3. Emission/observation probabilities (from $p(\mathbf{x}_n | \mathbf{z}_n)$)

If the observations are **discrete**, the emission probabilities B are a conditional probability table: $p(\mathbf{x}_t = l | z_t = k, \boldsymbol{\theta}) = B(k, l)$

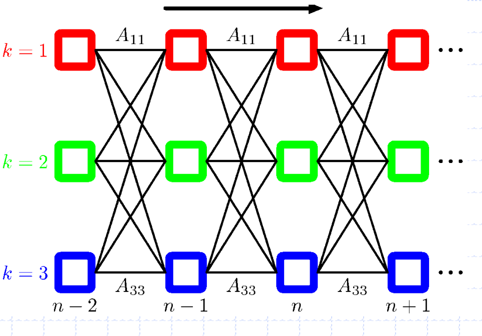
If the observations are **continuous**, $p(\mathbf{x}_n | \mathbf{z}_n)$ can be modelled by a Gaussian: $p(\mathbf{x}_t | z_t = k, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

(k is a state value index, e.g. z_{11} denotes the prior probability of latent variable z_1 assuming its first discrete value)

Hidden Markov Models (4)

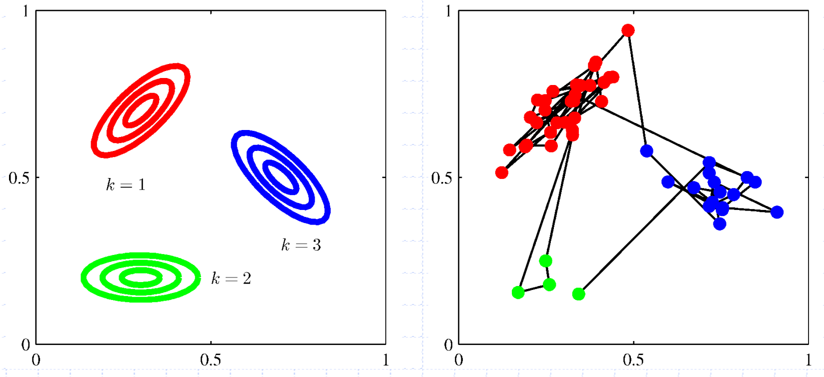


Transition diagram



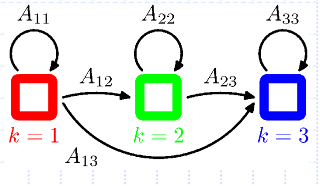
Lattice diagram

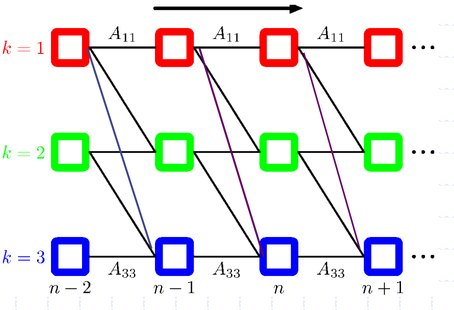
Hidden Markov Models (5)



Figures: Sampling from a 3-state HMM with a Gaussian emission probability and a 2-dimensional observation.

Hidden Markov Models (6)





Figures: a left-to-right HMM, typically used in online handwritten character recognition and speech recognition

Hidden Markov Models (7)

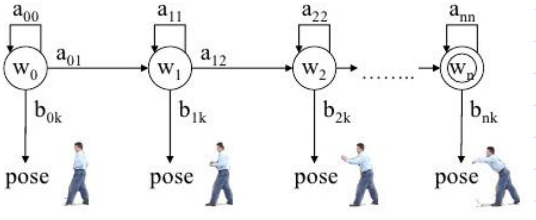
HMM applications

Automatic speech recognition: x = features extracted from the speech signal, z = words being spoken

Activity recognition: x = features extracted from the video frames, z = class of activity the person is engaged (e.g. walking)

Part of speech tagging: x = words, z = part of speech (e.g. noun)

Gene finding: x = DNA nucleotides (e.g. G), z = whether it is inside a gene-coding region or not.



Learning for HMMs (1)

Learning for HMMs:

How to estimate the parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ given observations

e.g. given a sequence of speech data, can we estimate transition and observation probabilities for words?

The most common approach is to use the EM algorithm - when applied to HMMs it is also called the **Baum-Welch algorithm.**

Expectation step:

$$\begin{aligned}\gamma(\mathbf{z}_n) &= p(\mathbf{z}_n | \mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \\ \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}, \boldsymbol{\theta}^{\text{old}})\end{aligned}$$

Learning for HMMs (2)

Maximization step:

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \qquad A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})}$$

Updating Gaussian emission densities:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \qquad \boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

Learning for HMMs (3)

But how to compute the posteriors in the expectation step?

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n | \mathbf{X}, \boldsymbol{\theta}^{\text{old}})$$
$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}, \boldsymbol{\theta}^{\text{old}})$$

There is an efficient procedure, in terms of O(K^2N), called the **forward-backward algorithm.**

The hidden state posterior can be expressed as a product of a "forward probability" with a "backward probability":

$$\gamma(\mathbf{z}_n) = \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_n) p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N | \mathbf{z}_n)}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

where

$$\alpha(\mathbf{z}_n) \equiv p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_n) \qquad \beta(\mathbf{z}_n) \equiv p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N | \mathbf{z}_n)$$

Learning for HMMs (4)

The forward and backward probabilities can be calculated recursively:

$$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1})$$

$$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) p(\mathbf{z}_{n+1}|\mathbf{z}_n)$$

And we can now update the posterior for the transitions:

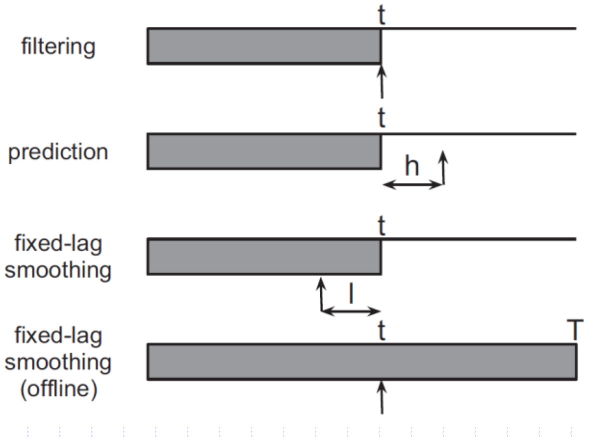
$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = \frac{\alpha(\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{z}_n|\mathbf{z}_{n-1}) \beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

...and this is how an HMM can be trained!

21

Inference for HMMs (1)

Inference for HMMs: how to infer a hidden state or a sequence of hidden states, assuming the HMM parameters are known



22

Inference for HMMs (2)

Prediction (for observations):

Let us assume that we have observed data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and we wish to predict the next observation, i.e. $\mathbf{x}_{N+1}$

This can be done using the forward probability:

$$p(\mathbf{x}_{N+1}|\mathbf{X}) = \frac{1}{p(\mathbf{X})} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N)\alpha(\mathbf{z}_N)$$

(used frequently in financial forecasting)

Inference for HMMs (3)

MAP estimation (Viterbi):

Let us assume that we have observed data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and we wish to estimate the most probable sequence of states:

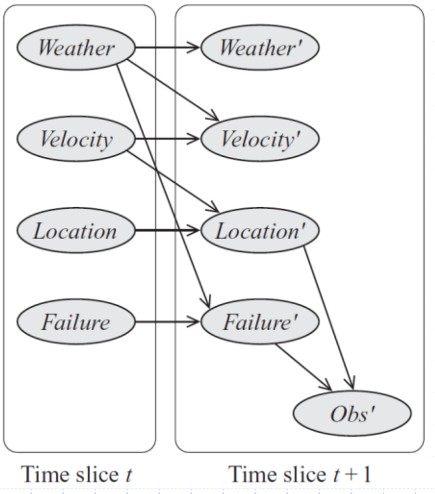
$$\mathbf{z}^* = \arg\max_{\mathbf{z}_{1:T}} p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$$

This problem can be solved efficiently using the **Viterbi algorithm.**

Note: *the (jointly) most probable sequence of states is not necessarily the same as the sequence of (individually) most probable states*

HMM Extensions…

Dynamic Bayesian Networks



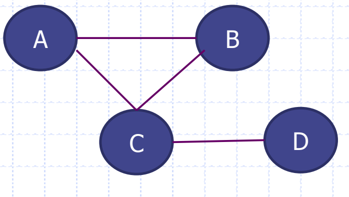
For more info, see K. Murphy's webpage/thesis:
"Dynamic Bayesian Networks: Representation, Inference and Learning"

http://www.cs.ubc.ca/~murphyk/Thesis/thesis.html

Conditional Random Fields

Undirected graphical model (i.e. based on a Markov network rather than a Bayesian network)

Markov network (undirected and possibly cyclic)



Related to Hopfield networks and Restricted Boltzmann Machines…

State Space Models (1)

A **state space model (SSM)** is just like an HMM, except the hidden states are continuous.

An SSM can be written in the following generic form:

$$\mathbf{z}_t = g(\mathbf{u}_t, \mathbf{z}_{t-1}, \boldsymbol{\epsilon}_t)$$
$$\mathbf{y}_t = h(\mathbf{z}_t, \mathbf{u}_t, \boldsymbol{\delta}_t)$$

- $\mathbf{z}_t$ is a hidden state
- $\mathbf{u}_t$ is an optional input or control signal
- $\mathbf{y}_t$ is the observation
- g is the transition model
- h is the observation/emission model
- $\boldsymbol{\varepsilon}_t$ is the system noise
- $\boldsymbol{\delta}_t$ is the observation noise

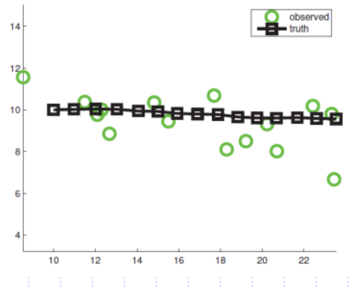
27

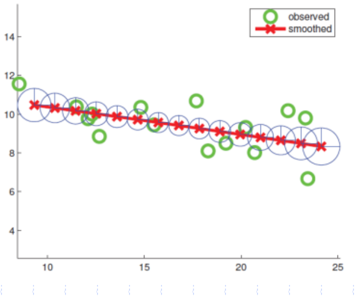
State Space Models (2)

An important special case of an SSM is where all the CPDs are Gaussian and the transition/observation models are linear functions. This is called a **linear dynamical system (LDS).**

Applications of SSMs:

- Object tracking
- Simultaneous localisation and mapping (SLAM) - robotics





28