COMS30035, Machine learning: Sequential Data 2: Hidden Markov Models

Edwin Simpson
edwin.simpson@bristol.ac.uk

Department of Computer Science, SCEEM
University of Bristol

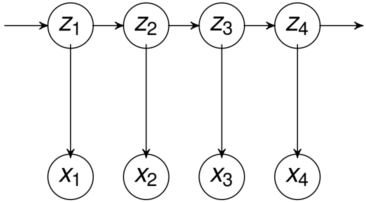
November 2, 2020

Agenda

- Markov Models
- Hidden Markov Models
- EM for HMMs
- Linear Dynamical Systems
- Bayesian Timeseries Modelling with Gaussian Processes

Hidden Markov Models (HMMs)

- A state space model
- z_i are latent (unobserved) discrete state variables.
- x_i are observations, which may be discrete or continuous values depending on the application.



Uses of HMMs: Sequence Labelling for Text

- *Sequence labelling*, i.e., classifying data points in a sequence.
- E.g., classifying words in a text document into grammatical categories such as "noun", "verb", "adjective", etc.
- This is called part-of-speech (POS) tagging and is used by natural language understanding systems, e.g., to extract facts and events from text data.

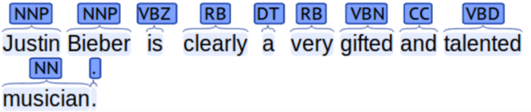
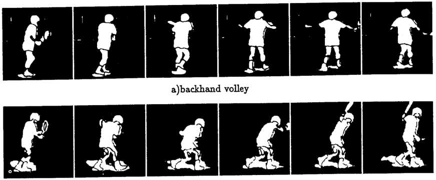


Image from "Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno", Yimam et al., 2014, ACL System Demonstrations.

Uses of HMMs: Human Action Recognition

- Observations: sequence of images (video frames) of a person playing tennis.
- Latent states: the actions being taken:
  - Backhand volley;
  - Forehand volley;
  - Forehand stroke;
  - Smash;
  - Serve.
- Why use an HMM? Actions typically follow a temporal sequence.



a)backhand volley

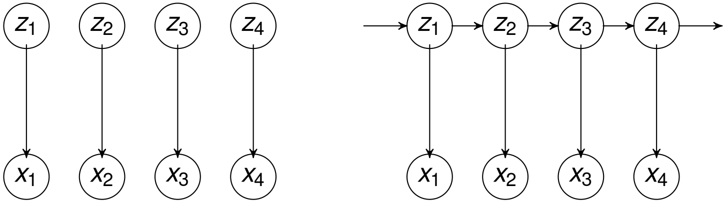
b)backhand stroke

Uses of HMMs: In General

- HMMs can be used with different goals in mind:
  - Inferring the latent states (sequence labelling);
  - Predicting the next latent state;
  - Predicting the next observation;
- They can also be used with different levels of supervision:
  - Supervised: the latent states are given in the training set.
  - Unsupervised: no labels are provided for the latent states, so the model seeks an assignment that best explains the observations given the model.
  - Semi-supervised: some labels are given, but the model is learned over both labelled and unlabelled sequences of data. This tries to avoid overfitting to a very small labelled dataset while identifying latent states that follow the desired labelling scheme.

HMM is an Extension to Mixture Models

- Recall the latent variables, z_i, in a mixture model, which identify the component responsible for an observation.
- These are also discrete variables, like latent states z_i in an HMM.
- In a mixture model, latent variables are i.i.d. rather than Markovian.



Transition Matrix

- The probability of a state z_i depends on the previous state: $p(z_i|z_{i-1})$.
- Given K labels (state values), we can write all the values of $p(z_i = k|z_{i-1} = l)$ in a *transition matrix*, $\boldsymbol{A}$.
  - Rows correspond to values of the previous state, z_{i-1}.
  - Columns are values of the current state, z_i.
  - Can you draw a transition matrix for a mixture model?
- Another vector of probabilities, π is used for z_1, since it has no predecessor.

$p(z_i\|z_{i-1}, \boldsymbol{A})$		z_i 1	2	3
z_{i-1}	1	0.5	0.1	0.4
	2	0.3	0.1	0.6
	3	0.01	0.19	0.8

Emission Probabilities

- Distribution over the observed variables, $p(x_i|z_i, \phi)$, where ϕ are parameters of the distributions, for example:
  - Real-valued observations may use Gaussian emission probabilities;
  - If there are multiple observations, we may use a multivariate Gaussian;
  - Discrete observations may use a categorical distribution.
- For each observation there are K values of $p(x_i|z_i, \phi)$, one for each possible value of the unobserved z_i.

The Complete HMM Model

- The complete HMM can be defined by the joint distribution over observations and latent states:

$$p(\boldsymbol{X}, \boldsymbol{Z} | \boldsymbol{A}, \boldsymbol{\pi}, \phi) = p(z_1 | \boldsymbol{\pi}) \prod_{n=2}^{N} p(z_n | z_{n-1}, \boldsymbol{A}) \prod_{n=1}^{N} p(x_n | z_n, \phi) \qquad (1)$$

- $\boldsymbol{A}$, ϕ and ϕ are parameters that must be learned or marginalised.
- Generative model: think of generating each of the state variables z_i in turn, then generating the observation x_i for each generated state.

Now do the quiz!

Please do the quiz for this lecture on Blackboard.
Next, we will see how to learn an HMM using the EM algorithm.