8: Hidden Markov Models

Machine Learning and Real-world Data

Simone Teufel and Ann Copestake

Computer Laboratory
University of Cambridge

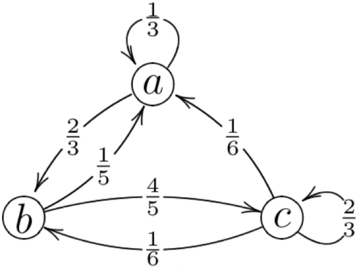
Lent 2017

Last session: catchup 1

- Research ideas from sentiment detection
- This concludes the part about statistical classification.
- We are now moving onto sequence learning.

Markov Chains

- A Markov Chain is a stochastic process with transitions from one state to another in a state space.
- Models sequential problems – your current situation depends on what happened in the past
- States are fully observable and discrete; transitions are labelled with transition probabilities.

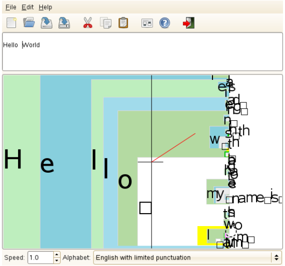


Markov Chains

- Once we observe a sequence of states, we can calculate a probability for a sequences of states we have been in.
- Important assumption: the probability distribution of the next state depends only on the current state
  - not on the sequence of events that preceded it.
- This model is appropriate in a number of applications, where states can be unambiguously observed.

Example: Predictive texting

- The famous A9 Algorithm, based on character n-grams
- A nice application based on it – Dasher, developed at Cambridge by David McKay



A harder problem

- But sometimes the observations are ambiguous with respect to their underlying causes
- In these cases, there is no 1:1 mapping between observations and states.
- A number of states can be associated with a particular observation, but the association of states and observations is governed by statistical behaviour.
- The states themselves are “hidden” from us.
- We only have access to the observations.
- We now have to *infer* the sequence of states that correspond to a sequence of observations.

Example where states are hidden

- Imagine a fraudulous croupier in a casino where customers bet on dice outcomes.
- She has two dice – a fair one and a loaded one.
- The fair one has the normal distribution of outcomes – $P(O) = \frac{1}{6}$ for each number 1 to 6.
- The loaded one has a different distribution.
- She secretly switches between the two dice.
- You don't know which dice is currently in use. You can only observe the numbers that are thrown.

Hidden Markov Model; States and Observations

$S_e = \{s_1, \ldots, s_N\}$	a set of N emitting states,
s_0	a special start state,
s_f	a special end state.

$K = \{k_1, \ldots k_m\}$	an output alphabet of M observations (vocabulary).

Hidden Markov Model; State and Observation Sequence

$O = o_1 \ldots o_T$	a sequence of T observations, each one drawn from K.
$X = X_1 \ldots X_T$	a sequence of T states, each one drawn from S_e.

Hidden Markov Model; State Transition Probabilities

A: a state transition probability matrix of size $(N+1) \times (N+1)$.

$$A = \begin{bmatrix} a_{01} & a_{02} & a_{03} & . & . & . & a_{0N} & - \\ a_{11} & a_{12} & a_{13} & . & . & . & a_{1N} & a_{1f} \\ a_{21} & a_{22} & a_{23} & . & . & . & a_{2N} & a_{2f} \\ . & . & . & & & & . & . \\ . & . & . & & & & . & . \\ . & . & . & & & & . & . \\ a_{N1} & a_{N2} & a_{N3} & . & . & . & a_{NN} & a_{Nf} \end{bmatrix}$$

a_{ij} is the probability of moving from state s_i to state s_j:

$$a_{ij} = P(X_t = s_j | X_{t-1} = s_i)$$

$$\forall_i \sum_{j=1}^{N} a_{ij} = 1$$

Hidden Markov Model; State Transition Probabilities

A: a state transition probability matrix of size $(N+1) \times (N+1)$.

$$A = \begin{bmatrix} a_{01} & a_{02} & a_{03} & . & . & . & a_{0N} & - \\ a_{11} & a_{12} & a_{13} & . & . & . & a_{1N} & a_{1f} \\ a_{21} & a_{22} & a_{23} & . & . & . & a_{2N} & a_{2f} \\ . & . & . & & & & . & . \\ . & . & . & & & & . & . \\ . & . & . & & & & . & . \\ a_{N1} & a_{N2} & a_{N3} & . & . & . & a_{NN} & a_{Nf} \end{bmatrix}$$

a_{ij} is the probability of moving from state s_i to state s_j:

$$a_{ij} = P(X_t = s_j | X_{t-1} = s_i)$$

$$\forall_i \sum_{j=1}^{N} a_{ij} = 1$$

Start state s_0 and end state s_f

- Not associated with observations
- a_{0i} describe transition probabilities out of the start state into state s_i
- a_{if} describe transition probabilities into the end state
- Transitions into start state (a_{i0}) and out of end state (a_{fi}) undefined.

Hidden Markov Model; Emission Probabilities

B: an emission probability matrix of size $N \times M$.

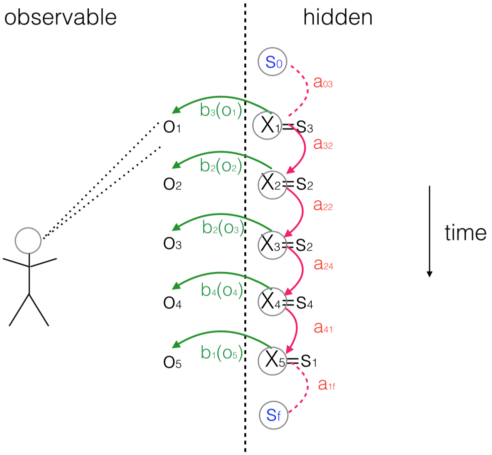
$$B = \begin{bmatrix} b_1(k_1) & b_2(k_1) & b_3(k_1) & . & . & . & b_N(k_1) \\ b_1(k_2) & b_2(k_2) & b_3(k_2) & . & . & . & b_N(k_2) \\ . & . & . & & & & . \\ . & . & . & & & & . \\ . & . & . & & & & . \\ b_1(k_M) & b_2(k_M) & b_3(k_M) & . & . & . & b_N(k_M) \end{bmatrix}$$

$b_i(k_j)$ is the probability of emitting vocabulary item k_j from state s_i:

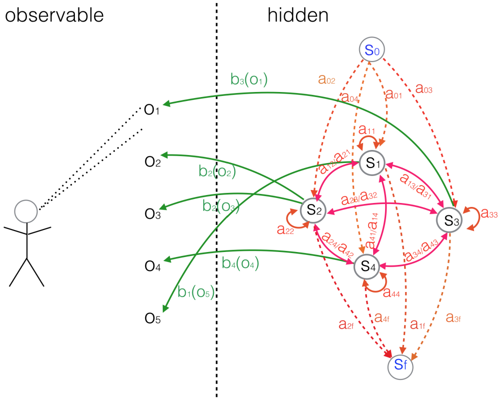
$$b_i(k_j) = P(O_t = k_j | X_t = s_i)$$

An HMM is defined by its parameters $\mu = (A, B)$.

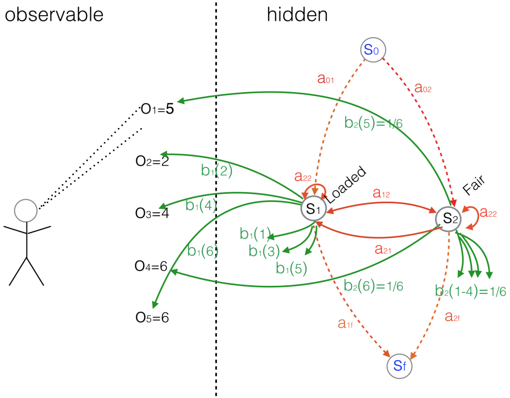
A Time-elapsed view of an HMM



A state-centric view of an HMM



The dice HMM



- There are two states (fair and loaded)
- Distribution of observations differs between the states

Markov assumptions

1. **Output Independence:** sequence of T observations. Each depends only on current state, not on history

$$P(O_t|X_1 ... X_t, ..., X_T, O_1, ..., O_t, ..., O_T) = P(O_t|X_t)$$

2. **Limited Horizon:** Transitions depend only on current state:

$$P(X_t|X_1 ... X_{t-1}) = P(X_t|X_{t-1})$$

- This is a first order HMM.
- In general, transitions in an HMM of order n depend on the past n states.

Tasks with HMMs

- Problem 1 (Labelled Learning)
  - Given a parallel observation and state sequence O and X, learn the HMM parameters A and B. $\rightarrow$ today
- Problem 2 (Unlabelled Learning)
  - Given an observation sequence O (and only the set of emitting states S_e), learn the HMM parameters A and B.
- Problem 3 (Likelihood)
  - Given an HMM $\mu = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\mu)$.
- Problem 4 (Decoding)
  - Given an observation sequence O and an HMM $\mu = (A, B)$, discover the best hidden state sequence X. $\rightarrow$ Task 8

Your Task today

Task 7:

- Your implementation performs labelled HMM learning, i.e. it has
  - Input: dual tape of state and observation (dice outcome) sequences X and O.

s_0	F	F	F	F	L	L	L	F	F	F	F	L	L	L	L	F	F	s_F
	1	3	4	5	6	6	5	1	2	3	1	4	3	5	4	1	2	

  - Output: HMM parameters A, B.
- As usual, the data is split into training, validation, test portions.
- Note: you will in a later task use your code for an HMM with more than two states. Either plan ahead now or modify your code later.

Parameter estimation of HMM parameters A, B

s_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	

- Transition matrix A consists of transition probabilities a_{ij}

$$a_{ij} = P(X_{t+1} = s_j | X_t = s_i) \sim \frac{count(X_t = s_i, X_{t+1} = s_j)}{count(X_t = s_i)}$$

- Emission matrix B consists of emission probabilities $b_i(k_j)$

$$b_i(k_j) = P(O_t = k_j | X_t = s_i) \sim \frac{count(O_t = k_j, X_t = s_i)}{count(X_t = s_i)}$$

- Add-one smoothed versions of these

Literature

- Manning and Schutze (2000). Foundations of Statistical Natural Language Processing, MIT Press. Chapters 9.1, 9.2.
  - We use state-emission HMM instead of arc-emission HMM
  - We avoid initial state probability vector π by using explicit start state s_0 and incorporating the corresponding probabilities into transition matrix A.
- (Jurafsky and Martin, 2nd Edition, Chapter 6.2 (but careful, notation!))