Hidden Markov Models

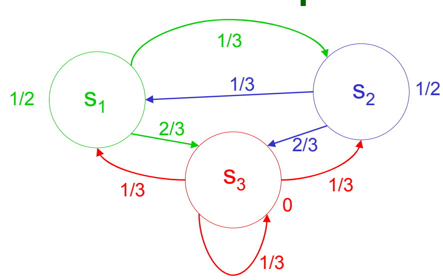
Ronald J. Williams

CSG220

Spring 2007

Contains several slides adapted from an Andrew Moore tutorial on this topic and a few figures from Russell & Norvig's *AIMA* site and Alpaydin's *Introduction to Machine Learning* site.

A Simple Markov Chain



Numbers at nodes represent probability of starting at the corresponding state.

Numbers on arcs represent transition probabilities.

At each time step, t = 1, 2, ... a new state is selected randomly according to the distribution at the current state.

Let X_t be a random variable for the state at time step t.

Let x_t represent the actual value of the state at time t. In this example, x_t can be s_1, s_2, or s_3.

Markov Property

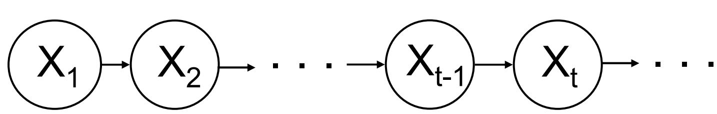
- For any t, X_{t+1} is conditionally independent of $\{X_{t-1}, X_{t-2}, \ldots X_1\}$ given X_t.
- In other words:

  $P(X_{t+1} = s_j | X_t = s_i) = P(X_{t+1} = s_j | X_t = s_i, \text{any earlier history})$
- Question: What would be the best Bayes Net structure to represent the Joint Distribution of $(X_1, X_2, \ldots, X_{t-1}, X_t, \ldots)$?

Markov Property

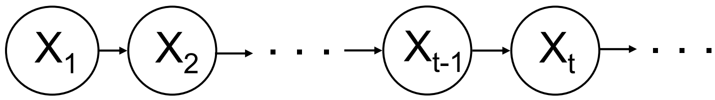
- For any t, X_{t+1} is conditionally independent of $\{X_{t-1}, X_{t-2}, \ldots X_1\}$ given X_t.
- In other words:

  $P(X_{t+1} = s_j | X_t = s_i) = P(X_{t+1} = s_j | X_t = s_i, \text{any earlier history})$
- Question: What would be the best Bayes Net structure to represent the Joint Distribution of $(X_1, X_2, \ldots, X_{t-1}, X_t, \ldots)$?
- Answer:



Markov chain as a Bayes net

i	$P(X_1 = s_i)$
1	π_1
2	π_2
3	π_3
i	π_i
N	π_N



	1	2	…	j	…	N
1	a_{11}	a_{12}	…	a_{1j}	…	a_{1N}
2	a_{21}	a_{22}	…	a_{2j}	…	a_{2N}
3	a_{31}	a_{32}	…	a_{3j}	…	a_{3N}
:	:	:	:	:	:	:
i	a_{i1}	a_{i2}	…	a_{ij}	…	a_{iN}
N	a_{N1}	a_{N2}	…	a_{Nj}	…	a_{NN}

Notation:
$$a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i)$$
$$\pi_i = P(X_1 = s_i)$$

Same CPT at every node except X_1

Markov Chain: Formal Definition

A Markov chain is a 3-tuple consisting of

- a set of N possible states $\{s_1, s_2, ..., s_N\}$
- $\{\pi_1, \pi_2, .. \pi_N\}$ The starting state probabilities
  $\pi_i = P(X_1 = s_i)$
- $$\left.\begin{matrix} a_{11} & a_{22} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ : & : & & : \\ a_{N1} & a_{N2} & \ldots & a_{NN} \end{matrix}\right\}$$
  The state transition probabilities
  $a_{ij} = P(X_{t+1}=s_j \mid X_t=s_i)$

Computing stuff in Markov chains

- Some notation and assumptions
  - Assume time t runs from 1 to T
  - Recall that X_t is the r.v. representing the state at time t and x_t denotes the actual value
  - Use $X_{t1:t2}$ and $x_{t1:t2}$ as shorthand for $(X_{t1}, X_{t1+1}, ..., X_{t2})$ and $(x_{t1}, x_{t1+1}, ... x_{t2})$, respectively
  - Use notation like $P(x_t)$ as shorthand for $P(X_t=x_t)$

What is $P(X_t = s_i)$? 1st attempt

Step 1: Work out how to compute $P(x_{1:t})$ for any state sequence $x_{1:t}$

$$
\begin{aligned}
P(x_{1:t}) &= P(x_t \mid x_{1:t-1})P(x_{1:t-1}) \\
&= P(x_t \mid x_{1:t-1})P(x_{t-1} \mid x_{1:t-2})P(x_{1:t-2}) \\
&\vdots \\
&= P(x_t \mid x_{1:t-1})P(x_{t-1} \mid x_{1:t-2}) \cdots P(x_2 \mid x_{1:1})P(x_{1:1}) \\
&= P(x_t \mid x_{t-1})P(x_{t-1} \mid x_{t-2}) \cdots P(x_2 \mid x_1)P(x_1)
\end{aligned}
$$

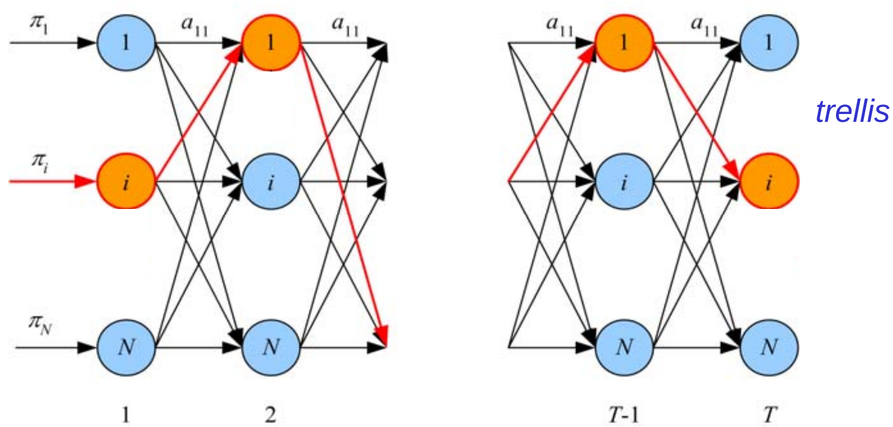
WHY?

Step 2: Use this knowledge to get $P(X_t = s_i)$

$$P(X_t = s_i) = \sum_{\text{sequences for which } x_t = s_i} P(x_{1:t})$$

Computation is exponential in t

State sequence as a path



Exponentially many paths, but at each time step only goes through exactly one of the N states

What is $P(X_t = s_i)$? Clever approach

- For each state s_i, define

$$p_t(i) = P(X_t = s_i)$$

- Express inductively

$$\forall i \quad p_1(i) \equiv P(X_1 = s_i) = \pi_i$$

$$\forall j \quad p_{t+1}(j) \equiv P(X_{t+1} = s_j)$$

$$= \sum_{i=1}^{N} P(X_{t+1} = s_j \mid X_t = s_i) P(X_t = s_i)$$

$$= \sum_{i=1}^{N} a_{ij} p_t(i)$$

What is $P(X_t = s_i)$? Clever approach

- For each state s_i, define

$$p_t(i) = P(X_t = s_i)$$

- Express inductively

$$\forall i \quad p_1(i) \equiv P(X_1 = s_i) = \pi_i$$

$$\forall j \quad p_{t+1}(j) \equiv P(X_{t+1} = s_j) = \sum_{i=1}^{N} P(X_{t+1} = s_j \mid X_t = s_i)P(X_t = s_i) = \sum_{i=1}^{N} a_{ij} p_t(i)$$

time step

state index	1	2	. . .	T
1				
2				
:				
N				

- Computation is simple.
- Just fill in this table one column at a time, from left to right
- Cells in this table correspond to nodes in the trellis

What is $P(X_t = s_i)$? Clever approach

- For each state s_i, define

$$p_t(i) = P(X_t = s_i)$$

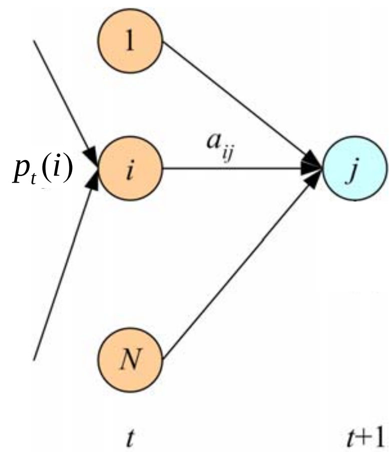
- Express inductively

$$\forall i \quad p_1(i) \equiv P(X_1 = s_i) = \pi_i$$

$$\forall j \quad p_{t+1}(j) \equiv P(X_{t+1} = s_j) = \sum_{i=1}^{N} P(X_{t+1} = s_j \mid X_t = s_i)P(X_t = s_i) = \sum_{i=1}^{N} a_{ij} p_t(i)$$

- Cost of computing $p_t(i)$ for all states s_i is now $O(TN^2)$
- The first way was $O(N^T)$
- This was a simple example
- It was meant to warm you up to this trick, called *Dynamic Programming,* because HMM computations involve many tricks just like this.

Inductive step: graphical representation



$$p_{t+1}(j) = \sum_{i=1}^{N} a_{ij} \, p_t(i)$$

Compare this with similar depictions of updates we'll use in HMMs

Hidden Markov Models: Slide 13

Hidden State

- Given a Markov model of a process, computation of various quantities of interest (e.g., probabilities) is straightforward if the state is observable – use techniques like the one just described.
- More realistic: assume the true state is not observable – only have observations that depend on, but do not fully determine, the actual states.
- Examples
  - Robot localization
    - state = actual location
    - observations = (noisy) sensor readings
  - Speech recognition
    - state sequence => word
    - observations = acoustic signal
- In this situation, we say the state is *hidden*
- Model this using a *Hidden Markov Model (HMM)*

Hidden Markov Models: Slide 14

HMMs

- An HMM is just a Markov chain augmented with
  - a set of M possible observations $\{o_1, o_2, ..., o_M\}$
  - for each state $s_1, s_2, ..., s_N$ a distribution over possible observations that might be sensed in that state
- We'll let Z_t be the r.v. for the observation that occurs at time t (with z_t representing the actual observation)
- In addition, we'll assume that the observation at time t depends *only* on the state at time t, in the sense about to be described

Markov Property of Observations

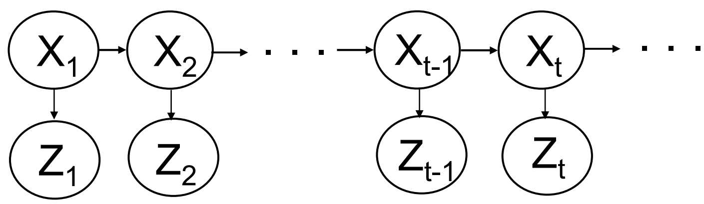
- For any t, Z_t is conditionally independent of $\{X_{t-1}, X_{t-2}, \ldots X_1, Z_{t-1}, Z_{t-2}, ..., Z_1\}$ given X_t.
- In other words:

  $P(Z_t = o_j | X_t = s_i) = P(Z_t = o_j | X_t = s_i$,any earlier history$)$
- Question: What would be the best Bayes Net structure to represent the Joint Distribution of $(X_1, Z_1, X_2, Z_2, \ldots, X_{t-1}, Z_{t-1}, X_t, Z_t, ...)$?

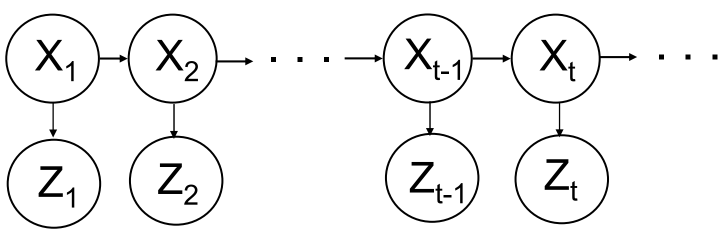
Markov Property of Observations

- For any t, Z_t is conditionally independent of $\{X_{t-1}, X_{t-2}, \ldots X_1, Z_{t-1}, Z_{t-2}, \ldots, Z_1\}$ given X_t.
- In other words:

  $P(Z_t = o_j \mid X_t = s_i) = P(Z_t = o_j \mid X_t = s_i$, any earlier history$)$
- Question: What would be the best Bayes Net structure to represent the Joint Distribution of $(X_1, Z_1, X_2, Z_2, \ldots, X_{t-1}, Z_{t-1}, X_t, Z_t, \ldots)$?
- Answer:



HMM as a Bayes Net



observation index

state index

	1	2	…	k	…	M
1	$b_1(o_1)$	$b_1(o_2)$	…	$b_1(o_k)$	…	$b_1(o_M)$
2	$b_2(o_1)$	$b_2(o_2)$	…	$b_2(o_k)$	…	$b_2(o_M)$
3	$b_3(o_1)$	$b_3(o_2)$	…	$b_3(o_k)$	…	$b_3(o_M)$
:	:	:	:	:	:	:
i	$b_i(o_1)$	$b_i(o_2)$	…	$b_i(o_k)$	…	$b_i(o_M)$
:	:	:	:	:	:	:
N	$b_N(o_1)$	$b_N(o_2)$	…	$b_N(o_k)$	…	$b_N(o_M)$

This is the CPT for every Z node

Notation:

$$b_i(o_k) = P(Z_t = o_k \mid X_t = s_i)$$

Are HMMs Useful?

You bet !!

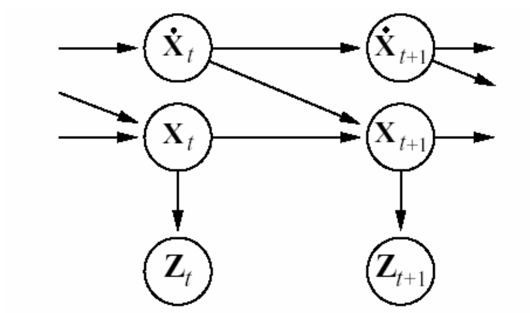
- Robot planning & sensing under uncertainty (e.g. Reid Simmons / Sebastian Thrun / Sven Koenig)
- Robot learning control (e.g. Yangsheng Xu's work)
- Speech Recognition/Understanding
  Phones $\rightarrow$ Words, Signal $\rightarrow$ phones
- Human Genome Project
  Complicated stuff your lecturer knows nothing about.
- Consumer decision modeling
- Economics & Finance.

Plus at least 5 other things I haven't thought of.

Dynamic Bayes Nets

- An HMM is actually a special case of a more general concept: Dynamic Bayes Net (DBN)
- Can decompose into multiple state variables and multiple observation variables at each time slice, with only direct influences represented explicitly
- (1^{st} order) Markov property: nodes in any time slice have arcs only from nodes in their own or the immediately preceding time slice
- Higher-order Markov models also easily represented in this framework

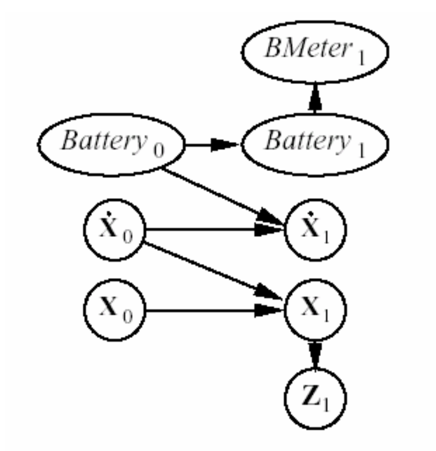
DBN Example



Linear dynamical system with position sensors

E.g., target tracking

Another DBN Example



Modeling a robot with position sensors and a battery charge meter

Back to HMMs ...

Summary of our HMM notation:

- X_t = state at time t (r.v.)
- Z_t = observation at time t (r.v.)
- $V_{t_1:t_2} = (V_{t_1}, V_{t_1+1}, ..., V_{t_2})$ for any time-indexed r.v. V
- Possible states = $\{s_1, s_2, ..., s_N\}$
- Possible observations = $\{o_1, o_2, ..., o_M\}$
- v_t = actual value of r.v. V at time step t
- $v_{t_1:t_2} = (v_{t_1}, v_{t_1+1}, ..., v_{t_2})$ = sequence of actual values of r.v. V from time steps t_1 through t_2
- Convenient shorthand: E.g., $P(x_{1:t} \mid z_{1:t})$ means $P(X_{1:t} = x_{1:t} \mid Z_{1:t} = z_{1:t})$
- T = final time step

Hidden Markov Models: Slide 23

HMM: Formal Definition

An HMM λ is a 5-tuple consisting of

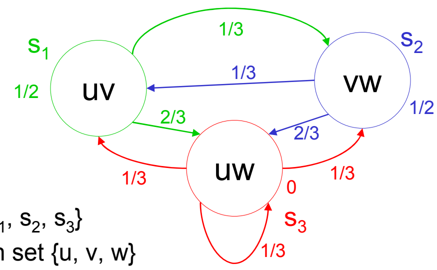
- a set of N possible states $\{s_1, s_2, ..., s_N\}$
- a set of M possible observations $\{o_1, o_2, ..., o_M\}$
- $\{\pi_1, \pi_2, .. \pi_N\}$ The starting state probabilities
  $\pi_i = P(X_1 = s_i)$
- $$\left.\begin{matrix} a_{11} & a_{22} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ : & : & & : \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{matrix}\right\}$$ The state transition probabilities
  $a_{ij} = P(X_{t+1}=s_j \mid X_t=s_i)$
- $$\left.\begin{matrix} b_1(o_1) & b_1(o_2) & \dots & b_1(o_M) \\ b_2(o_1) & b_2(o_2) & \dots & b_2(o_M) \\ : & : & & : \\ b_N(o_1) & b_N(o_2) & \dots & b_N(o_M) \end{matrix}\right\}$$ The observation probabilities
  $b_i(o_k) = P(Z_t=o_k \mid X_t=s_i)$

Hidden Markov Models: Slide 24

Here's an HMM

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.



State set $\{s_1, s_2, s_3\}$
Observation set $\{u, v, w\}$

$\pi_1 = 1/2$ $\quad$ $\pi_2 = 1/2$ $\quad$ $\pi_3 = 0$

$a_{11} = 0$ $\quad$ $a_{12} = 1/3$ $\quad$ $a_{13} = 2/3$
$a_{12} = 1/3$ $\quad$ $a_{22} = 0$ $\quad$ $a_{13} = 2/3$
$a_{13} = 1/3$ $\quad$ $a_{32} = 1/3$ $\quad$ $a_{13} = 1/3$

$b_1(u) = 1/2$ $\quad$ $b_1(v) = 1/2$ $\quad$ $b_1(w) = 0$
$b_2(u) = 0$ $\quad$ $b_2(v) = 1/2$ $\quad$ $b_2(w) = 1/2$
$b_3(u) = 1/2$ $\quad$ $b_3(v) = 0$ $\quad$ $b_3(w) = 1/2$

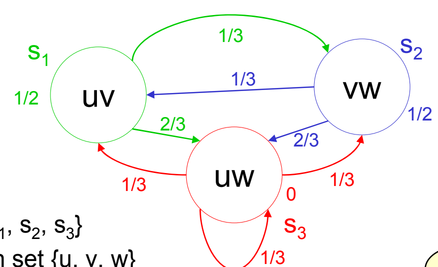
Hidden Markov Models: Slide 25

Here's an HMM

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

Let's generate a sequence of observations:



50-50 choice between s_1 and s_2

x_1=	___	z_1=	___
x_2=	___	z_2=	___
x_3=	___	z_3=	___

State set $\{s_1, s_2, s_3\}$
Observation set $\{u, v, w\}$

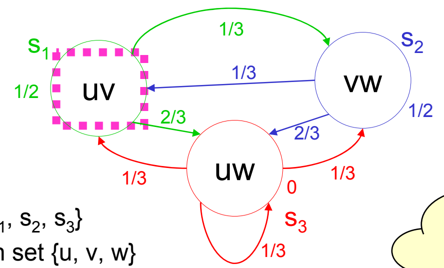
$\pi_1 = 1/2$ $\quad$ $\pi_2 = 1/2$ $\quad$ $\pi_3 = 0$

$a_{11} = 0$ $\quad$ $a_{12} = 1/3$ $\quad$ $a_{13} = 2/3$
$a_{12} = 1/3$ $\quad$ $a_{22} = 0$ $\quad$ $a_{13} = 2/3$
$a_{13} = 1/3$ $\quad$ $a_{32} = 1/3$ $\quad$ $a_{13} = 1/3$

$b_1(u) = 1/2$ $\quad$ $b_1(v) = 1/2$ $\quad$ $b_1(w) = 0$
$b_2(u) = 0$ $\quad$ $b_2(v) = 1/2$ $\quad$ $b_2(w) = 1/2$
$b_3(u) = 1/2$ $\quad$ $b_3(v) = 0$ $\quad$ $b_3(w) = 1/2$

Hidden Markov Models: Slide 26

Here's an HMM



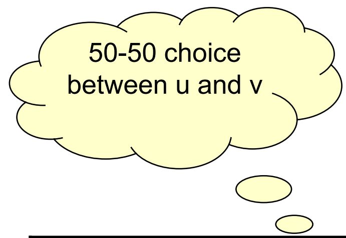
State set $\{s_1, s_2, s_3\}$
Observation set {u, v, w}

π_1 = 1/2	π_2 = 1/2	π_3 = 0
a_{11} = 0	a_{12} = 1/3	a_{13} = 2/3
a_{12} = 1/3	a_{22} = 0	a_{13} = 2/3
a_{13} = 1/3	a_{32} = 1/3	a_{13} = 1/3
b_1 (u) = 1/2	b_1 (v) = 1/2	b_1 (w) = 0
b_2 (u) = 0	b_2 (v) = 1/2	b_2 (w) = 1/2
b_3 (u) = 1/2	b_3 (v) = 0	b_3 (w) = 1/2

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

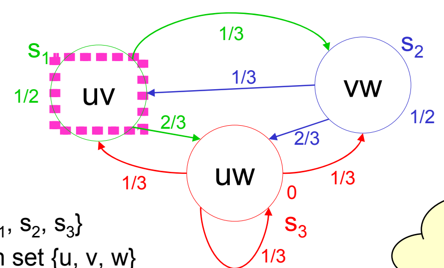
Let's generate a sequence of observations:



x_1=	s_1	z_1=	__
x_2=	__	z_2=	__
x_3=	__	z_3=	__

Hidden Markov Models: Slide 27

Here's an HMM



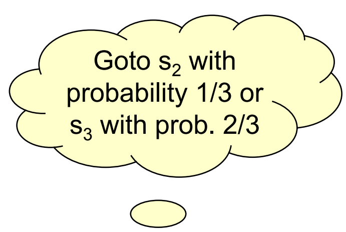
State set $\{s_1, s_2, s_3\}$
Observation set {u, v, w}

π_1 = 1/2	π_2 = 1/2	π_3 = 0
a_{11} = 0	a_{12} = 1/3	a_{13} = 2/3
a_{12} = 1/3	a_{22} = 0	a_{13} = 2/3
a_{13} = 1/3	a_{32} = 1/3	a_{13} = 1/3
b_1 (u) = 1/2	b_1 (v) = 1/2	b_1 (w) = 0
b_2 (u) = 0	b_2 (v) = 1/2	b_2 (w) = 1/2
b_3 (u) = 1/2	b_3 (v) = 0	b_3 (w) = 1/2

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

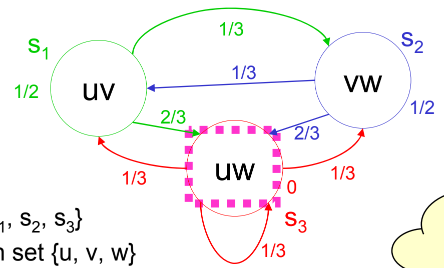
Let's generate a sequence of observations:



x_1=	s_1	z_1=	u
x_2=	__	z_2=	__
x_3=	__	z_3=	__

Hidden Markov Models: Slide 28

Here's an HMM



State set $\{s_1, s_2, s_3\}$
Observation set {u, v, w}

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$
$a_{12} = 1/3$ $a_{22} = 0$ $a_{13} = 2/3$
$a_{13} = 1/3$ $a_{32} = 1/3$ $a_{13} = 1/3$

$b_1(u) = 1/2$ $b_1(v) = 1/2$ $b_1(w) = 0$
$b_2(u) = 0$ $b_2(v) = 1/2$ $b_2(w) = 1/2$
$b_3(u) = 1/2$ $b_3(v) = 0$ $b_3(w) = 1/2$

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

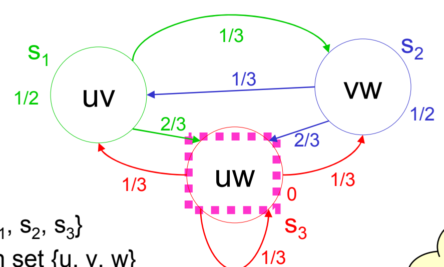
Let's generate a sequence of observations:

50-50 choice between u and w

x_1=	s_1	z_1=	u
x_2=	s_3	z_2=	__
x_3=	__	z_3=	__

Hidden Markov Models: Slide 29

Here's an HMM



State set $\{s_1, s_2, s_3\}$
Observation set {u, v, w}

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$
$a_{12} = 1/3$ $a_{22} = 0$ $a_{13} = 2/3$
$a_{13} = 1/3$ $a_{32} = 1/3$ $a_{13} = 1/3$

$b_1(u) = 1/2$ $b_1(v) = 1/2$ $b_1(w) = 0$
$b_2(u) = 0$ $b_2(v) = 1/2$ $b_2(w) = 1/2$
$b_3(u) = 1/2$ $b_3(v) = 0$ $b_3(w) = 1/2$

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

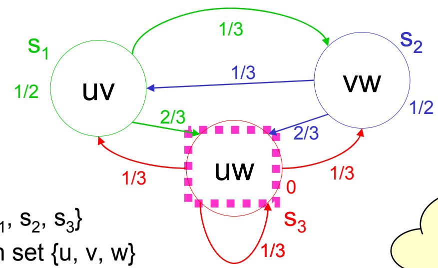
Let's generate a sequence of observations:

Each of the three next states is equally likely

x_1=	s_1	z_1=	u
x_2=	s_3	z_2=	u
x_3=	__	z_3=	__

Hidden Markov Models: Slide 30

Here's an HMM



State set $\{s_1, s_2, s_3\}$
Observation set {u, v, w}

π_1 = 1/2 $\qquad$ π_2 = 1/2 $\qquad$ π_3 = 0

a_{11} = 0	a_{12} = 1/3	a_{13} = 2/3
a_{12} = 1/3	a_{22} = 0	a_{13} = 2/3
a_{13} = 1/3	a_{32} = 1/3	a_{13} = 1/3

b_1 (u) = 1/2	b_1 (v) = 1/2	b_1 (w) = 0
b_2 (u) = 0	b_2 (v) = 1/2	b_2 (w) = 1/2
b_3 (u) = 1/2	b_3 (v) = 0	b_3 (w) = 1/2

Start randomly in state 1 or 2

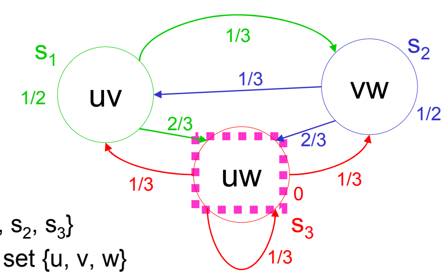
Choose one of the output symbols in each state at random.

Let's generate a sequence of observations:

50-50 choice between u and w

x_1=	s_1	z_1=	u
x_2=	s_3	z_2=	u
x_3=	s_3	z_3=	___

Here's an HMM



State set $\{s_1, s_2, s_3\}$
Observation set {u, v, w}

π_1 = 1/2 $\qquad$ π_2 = 1/2 $\qquad$ π_3 = 0

a_{11} = 0	a_{12} = 1/3	a_{13} = 2/3
a_{12} = 1/3	a_{22} = 0	a_{13} = 2/3
a_{13} = 1/3	a_{32} = 1/3	a_{13} = 1/3

b_1 (u) = 1/2	b_1 (v) = 1/2	b_1 (w) = 0
b_2 (u) = 0	b_2 (v) = 1/2	b_2 (w) = 1/2
b_3 (u) = 1/2	b_3 (v) = 0	b_3 (w) = 1/2

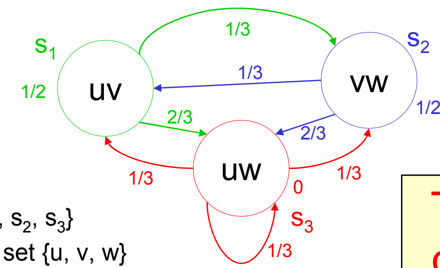
Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

Let's generate a sequence of observations:

x_1=	s_1	z_1=	u
x_2=	s_3	z_2=	u
x_3=	s_3	z_3=	w

Hidden State



Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

Let's generate a sequence of observations:

This is what the observer has to work with…

State set $\{s_1, s_2, s_3\}$
Observation set $\{u, v, w\}$

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$
$a_{12} = 1/3$ $a_{22} = 0$ $a_{13} = 2/3$
$a_{13} = 1/3$ $a_{32} = 1/3$ $a_{13} = 1/3$

b_1 (u) = 1/2 b_1 (v) = 1/2 b_1 (w) = 0
b_2 (u) = 0 b_2 (v) = 1/2 b_2 (w) = 1/2
b_3 (u) = 1/2 b_3 (v) = 0 b_3 (w) = 1/2

x_1=	?	z_1=	u
x_2=	?	z_2=	u
x_3=	?	z_3=	w

Problems to solve

- So now we have an HMM (or, more generally, a DBN) that models a temporal process of interest
- What are some of the kinds of problems we'd like to be able to solve with this?

Temporal Model Problems to Solve

- Filtering: Compute $P(X_t | z_{1:t}, \lambda)$
- Prediction: Compute $P(X_k | z_{1:t}, \lambda)$ for $k > t$
- Smoothing: Compute $P(X_k | z_{1:t}, \lambda)$ for $k < t$
- Observation sequence likelihood: Compute $P(z_{1:T} | \lambda)$
- Most probable path (state sequence): Compute $x_{1:T}$ maximizing $P(x_{1:T} | z_{1:T}, \lambda)$
- Maximum likelihood model: Given a set of observation sequences $\left\{ z^r_{1:T_r} \right\}_r$, compute λ maximizing $\prod_r P(z^r_{1:T_r} | \lambda)$

Hidden Markov Models: Slide 35

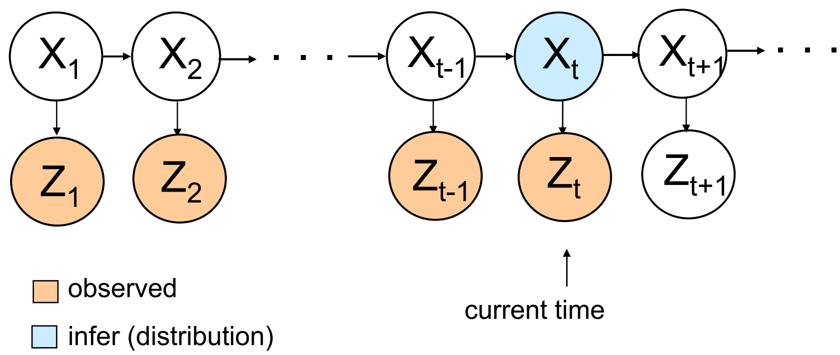
Temporal Model Problems to Solve

- Used in a wide variety of dynamical systems modeling applications:
  - filtering
  - prediction
  - smoothing
- Used especially in HMM applications:
  - observation sequence likelihood
  - most probable path
  - maximum likelihood model fitting

Hidden Markov Models: Slide 36

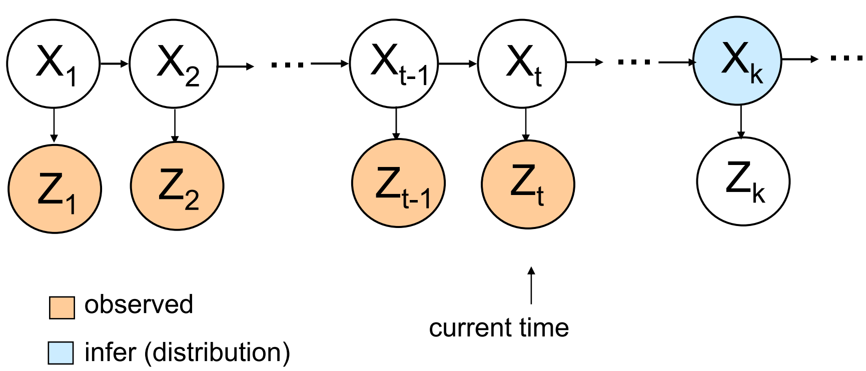
Filtering

Compute $P\left(X_t \middle| z_{1:t}, \lambda\right)$



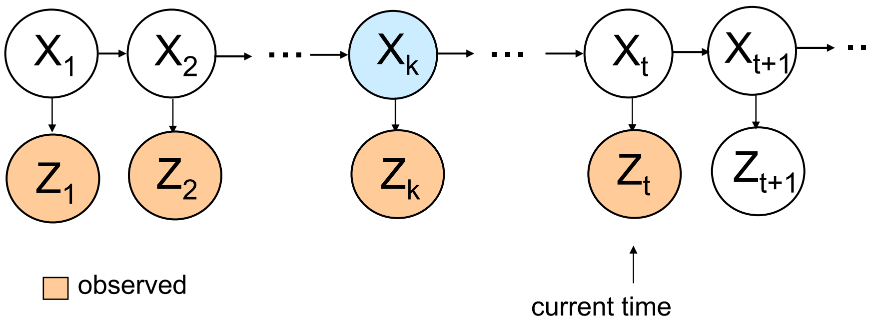
Prediction

Compute $P\left(X_k \middle| z_{1:t}, \lambda\right)$ for $k > t$



Smoothing

Compute $P(X_k | z_{1:t}, \lambda)$ for $k < t$



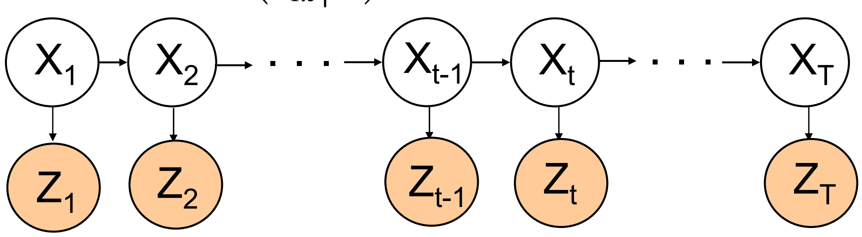
observed

infer (distribution)

Hidden Markov Models: Slide 39

Observation Sequence Likelihood

Compute $P(z_{1:t} | \lambda)$



observed

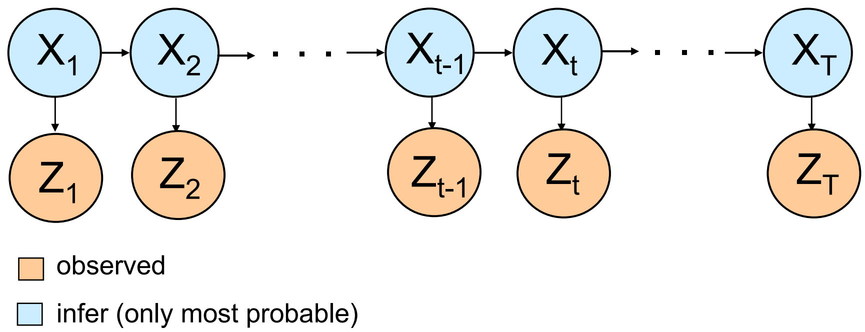
What's the probability of this particular sequence of observations as a function of the model parameters?

Useful for such things as finding which of a set of HMM models best fits an observation sequence, as in speech recognition.

Hidden Markov Models: Slide 40

Most Probable Path

Compute $\arg\max_{x_{1:T}} P\left(x_{1:T} \middle| z_{1:T}, \lambda\right)$



Not necessarily the same as the sequence of individually most probable states (obtained by smoothing)

Maximum Likelihood Model

Assume number of states given

Given a set of *R* observation sequences

$$
\begin{aligned}
z^1_{1:T_1} &= \left(z^1_1, z^1_2, \ldots, z^1_{T_1}\right) \\
z^2_{1:T_2} &= \left(z^2_1, z^2_2, \ldots, z^2_{T_2}\right) \\
&\vdots \\
z^R_{1:T_R} &= \left(z^R_1, z^R_2, \ldots, z^R_{T_R}\right)
\end{aligned}
$$

Compute

$$\lambda^* = \arg\max_{\lambda} \prod_{r=1}^{R} P(z^r_{1:T_r} \mid \lambda)$$

Solution methods for these problems

Let's start by considering the observation sequence likelihood problem:

$$\boxed{\text{Given } z_{1:T}\text{, compute } P\left(z_{1:t} \middle| \lambda\right)}$$

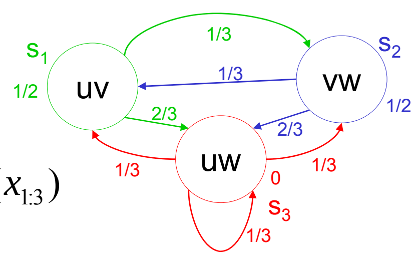
Use our example HMM to illustrate

Hidden Markov Models: Slide 43

Prob. of a sequence of 3 observations

$$P(z_{1:3}) = \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \wedge x_{1:3})$$

$$= \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \mid x_{1:3}) P(x_{1:3})$$



How do we compute $P(x_{1:3})$ for an arbitrary path $x_{1:3}$?

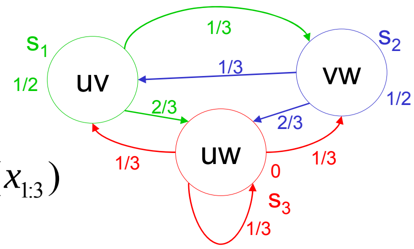
How do we compute $P(z_{1:3}|x_{1:3})$ for an arbitrary path $x_{1:3}$?

Hidden Markov Models: Slide 44

Prob. of a sequence of 3 observations

$$P(z_{1:3}) = \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \wedge x_{1:3})$$

$$= \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \mid x_{1:3}) P(x_{1:3})$$



How do we compute $P(x_{1:3})$ for an arbitrary path $x_{1:3}$?

$P(x_1,x_2,x_3) = P(x_1)\, P(x_2|x_1)\, P(x_3| x_2)$

E.g, $P(s_1, s_3, s_3) = 1/2 * 2/3 * 1/3 = 1/9$

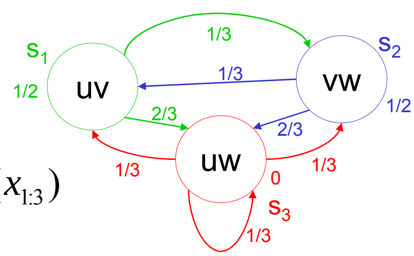
How do we compute $P(z_{1:3}|x_{1:3})$ for an arbitrary path $x_{1:3}$?

Hidden Markov Models: Slide 45

Prob. of a sequence of 3 observations

$$P(z_{1:3}) = \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \wedge x_{1:3})$$

$$= \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \mid x_{1:3}) P(x_{1:3})$$



How do we compute $P(x_{1:3})$ for an arbitrary path $x_{1:3}$?

$P(x_1,x_2,x_3) = P(x_1)\, P(x_2|x_1)\, P(x_3| x_2)$

E.g, $P(s_1, s_3, s_3) = 1/2 * 2/3 * 1/3 = 1/9$

How do we compute $P(z_{1:3}|x_{1:3})$ for an arbitrary path $x_{1:3}$?

$P(z_1, z_2, z_3 \mid x_1, x_2, x_3)$

$= P(z_1 \mid x_1)\, P(z_2 \mid x_2)\, P(z_3 \mid x_3)$

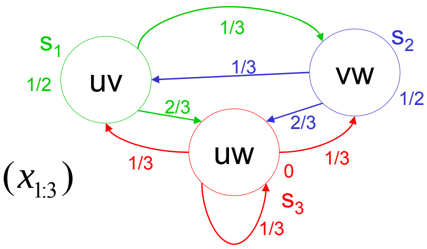
E.g, $P(uuw \mid s_1, s_3, s_3) = 1/2 * 1/2 * 1/2 = 1/8$

Hidden Markov Models: Slide 46

Prob. of a sequence of 3 observations

$$P(z_{1:3}) = \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \wedge x_{1:3})$$

$$= \sum_{x_{1:3} \in \text{paths of length 3}} P(z_{1:3} \mid x_{1:3}) P(x_{1:3})$$



But this sum has $3^3 = 27$ terms in it!

Exponential in the length of the sequence

Need to use a dynamic programming trick like before

The probability of a given sequence of observations, non-exponential-cost-style

Given observation sequence $(z_1, z_2, \ldots, z_T) = z_{1:T}$

Define the *forward variable*

$\alpha_t(i) = P(z_{1:t}, X_t = s_i \mid \lambda)$ for $1 \le t \le T$

$\alpha_t(i)$ = Probability that, in a random trial,

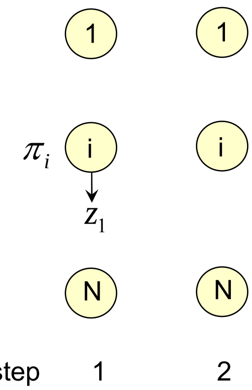
- we'd have seen the first t observations; and
- we'd have ended up in s_i as the t^{th} state visited.

Computing the forward variables

Base case:

$$\alpha_1(i) \equiv P(z_1 \wedge X_1 = s_i)$$
$$= P(z_1 | X_1 = s_i) P(X_1 = s_i)$$
$$= b_i(z_1)\pi_i$$

Note: For simplicity, we'll drop explicit reference to conditioning on the HMM parameters λ for many of the upcoming slides, but it's always there implicitly.



Forward variables: inductive step

$$\alpha_{t+1}(j) \equiv P(z_{1:t+1} \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$

sum over all possible previous states

Forward variables: inductive step

$$\alpha_{t+1}(j) \equiv P(z_{1:t+1} \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t} \wedge z_{t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$

split off last observation

Forward variables: inductive step

$$\alpha_{t+1}(j) \equiv P(z_{1:t+1} \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t} \wedge z_{t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | z_{1:t} \wedge X_t = s_i) P(z_{1:t} \wedge X_t = s_i)$$

chain rule

Forward variables: inductive step

$$\alpha_{t+1}(j) \equiv P(z_{1:t+1} \wedge X_{t+1} = s_j)$$

$$= \sum_{i=1}^{N} P(z_{1:t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$

$$= \sum_{i=1}^{N} P(z_{1:t} \wedge z_{t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$

$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | z_{1:t} \wedge X_t = s_i) P(z_{1:t} \wedge X_t = s_i)$$

$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | X_t = s_i) \alpha_t(i)$$

latest state and observation conditionally independent of earlier observations given previous state

Hidden Markov Models: Slide 53

Forward variables: inductive step

$$\alpha_{t+1}(j) \equiv P(z_{1:t+1} \wedge X_{t+1} = s_j)$$

$$= \sum_{i=1}^{N} P(z_{1:t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$

$$= \sum_{i=1}^{N} P(z_{1:t} \wedge z_{t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$

$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | z_{1:t} \wedge X_t = s_i) P(z_{1:t} \wedge X_t = s_i)$$

$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | X_t = s_i) \alpha_t(i)$$

$$= \sum_{i=1}^{N} P(z_{t+1} | X_{t+1} = s_j \wedge X_t = s_i) P(X_{t+1} = s_j | X_t = s_i) \alpha_t(i)$$

chain rule

Hidden Markov Models: Slide 54

Forward variables: inductive step

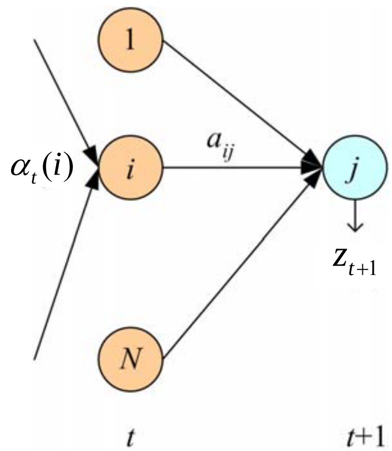
$$\alpha_{t+1}(j) \equiv P(z_{1:t+1} \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t} \wedge z_{t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | z_{1:t} \wedge X_t = s_i) P(z_{1:t} \wedge X_t = s_i)$$
$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | X_t = s_i) \alpha_t(i)$$
$$= \sum_{i=1}^{N} P(z_{t+1} | X_{t+1} = s_j \wedge X_t = s_i) P(X_{t+1} = s_j | X_t = s_i) \alpha_t(i)$$
$$= \sum_{i=1}^{N} P(z_{t+1} | X_{t+1} = s_j) a_{ij} \alpha_t(i)$$

latest observation conditionally independent of earlier states given latest state

Forward variables: inductive step

$$\alpha_{t+1}(j) \equiv P(z_{1:t+1} \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{1:t} \wedge z_{t+1} \wedge X_t = s_i \wedge X_{t+1} = s_j)$$
$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | z_{1:t} \wedge X_t = s_i) P(z_{1:t} \wedge X_t = s_i)$$
$$= \sum_{i=1}^{N} P(z_{t+1} \wedge X_{t+1} = s_j | X_t = s_i) \alpha_t(i)$$
$$= \sum_{i=1}^{N} P(z_{t+1} | X_{t+1} = s_j \wedge X_t = s_i) P(X_{t+1} = s_j | X_t = s_i) \alpha_t(i)$$
$$= \sum_{i=1}^{N} P(z_{t+1} | X_{t+1} = s_j) a_{ij} \alpha_t(i)$$
$$= b_j(z_{t+1}) \sum_{i=1}^{N} a_{ij} \alpha_t(i)$$

Forward variables: inductive step



$$\alpha_{t+1}(j) = b_j(z_{t+1}) \sum_{i=1}^{N} a_{ij}\alpha_t(i)$$

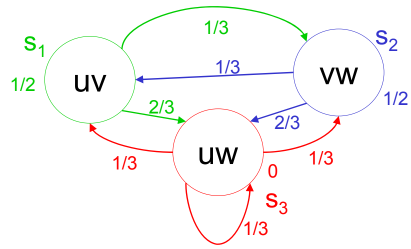
Observation Sequence Likelihood

Efficient solution to the *observation sequence likelihood* problem using the forward variables:

$$P(z_{1:t} | \lambda) = \sum_{i=1}^{N} P(z_{1:t} \wedge X_t = s_i | \lambda) = \sum_{i=1}^{N} \alpha_t(i)$$

In our example

$$\alpha_t(i) \equiv \mathrm{P}(z_{1:t} \wedge X_t = s_i | \lambda)$$
$$\alpha_1(i) = b_i(z_1)\pi_i$$
$$\alpha_{t+1}(j) = b_j(z_{t+1})\sum_i a_{ij}\alpha_t(i)$$



Observed: $z_1\ z_2\ z_3$ = u u w

$$\alpha_1(1) = \tfrac{1}{4} \qquad \alpha_2(1) = 0 \qquad \alpha_2(1) = 0$$
$$\alpha_1(2) = 0 \qquad \alpha_2(2) = 0 \qquad \alpha_3(2) = \tfrac{1}{72}$$
$$\alpha_1(3) = 0 \qquad \alpha_2(3) = \tfrac{1}{12} \qquad \alpha_3(3) = \tfrac{1}{72}$$

So probability of observing uuw is 1/36

Filtering

Efficient solution to the *filtering* problem using the forward variables:

$$P(X_t = s_i | z_{1:t}) = \frac{P(X_t = s_i \wedge z_{1:t})}{P(z_{1:t})} = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}$$

Estimating current state based on all observations up to the current time.

So in our example, after observing uuw, prob. of being in s_1 is 0 and prob. of being in s_2 = prob. of being in s_3 = 1/2

Prediction

- Note that the (state) prediction problem can be viewed as a special case of the filtering problem in which there are missing observations.
- That is, trying to compute the probability of X_k given observations up through time step t, with k > t, amounts to filtering with missing observations at time steps t+1, t+2, ..., k.
- Therefore, we now focus on the *missing observations problem*.

Missing Observations

- Looking at the derivation of the inductive step for computing the forward variables, we see that the last step involves writing

$$\alpha_{t+1}(j) = \underbrace{P(z_{t+1} | X_{t+1} = s_j)}_{b_j(z_{t+1})} \underbrace{P(X_{t+1} = s_j | \text{all observations up through time } t)}_{\sum_{i=1}^{N} a_{ij}\alpha_t(i)}$$

- Thus the second factor gives us a prediction of the state at time t+1 based on all earlier observations, which we then multiply by the observation probability at time t+1 given the state at time t+1.
- If there is no observation at time t+1, clearly the set of observations made through time t+1 is the same as the set of observations made through time t.

Missing Observations (cont.)

- Thus we redefine

$$\alpha_t(i) = P(X_t = s_i \wedge \text{all available observations through time } t)$$

- This generalizes our earlier definition but allows for the possibility that some observations are present and others are missing
- Then define

$$b'_i(z_t) = \begin{cases} b_i(z_t) & \text{if there is an observation at time } t \\ 1 & \text{otherwise} \end{cases}$$

- It's not hard to see that the correct forward compution should then proceed as:

$$\alpha_1(i) = b'_i(z_1)\pi_i$$
$$\alpha_{t+1}(j) = b'_j(z_{t+1})\sum_i a_{ij}\alpha_t(i)$$

- Amounts to propagating state predictions forward wherever there are no observations
- Interesting special case: When there are *no* observations at any time, the α values are identical to the *p* values we defined earlier for Markov chains

Solving the smoothing problem

- Define the *backward variables*

$$\beta_t(i) = P(z_{t+1:T} | X_t = s_i, \lambda)$$

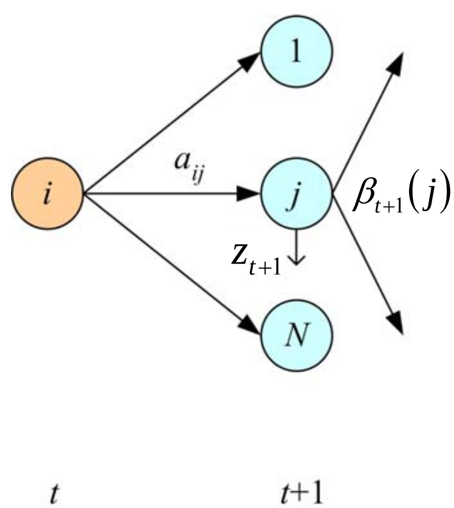
- Probability of observing z_{t+1}, ..., z_T given that system was in state s_i at time step t
- These can be computed efficiently by starting at the end (time T) and working backwards
- Base case: $\beta_T(i) = 1 \quad \text{for all } i,\ 1 \le i \le N$
  - Valid because $z_{T+1:T}$ is an empty sequence of observations so its probability is 1

Backward variables: inductive step

$$
\begin{aligned}
\beta_t(i) &\equiv P\left(z_{t+1:T} \middle| X_t = s_i\right) \\
&= \sum_{j=1}^{N} P\left(z_{t+1:T} \wedge X_{t+1} = s_j \middle| X_t = s_i\right) \\
&= \sum_{j=1}^{N} P\left(z_{t+1:T} \middle| X_{t+1} = s_j \wedge X_t = s_i\right) P\left(X_{t+1} = s_j \middle| X_t = s_i\right) \\
&= \sum_{j=1}^{N} P\left(z_{t+1} \wedge z_{t+2:T} \middle| X_{t+1} = s_j \wedge X_t = s_i\right) a_{ij} \\
&= \sum_{j=1}^{N} P\left(z_{t+1} \wedge z_{t+2:T} \middle| X_{t+1} = s_j\right) a_{ij} \\
&= \sum_{j=1}^{N} P\left(z_{t+1} \middle| z_{t+2:T} \wedge X_{t+1} = s_j\right) P\left(z_{t+2:T} \middle| X_{t+1} = s_j\right) a_{ij} \\
&= \sum_{j=1}^{N} P\left(z_{t+1} \middle| X_{t+1} = s_j\right) \beta_{t+1}(j) a_{ij} \\
&= \sum_{j=1}^{N} b_j(z_{t+1}) \beta_{t+1} a_{ij}
\end{aligned}
$$

Backward variables: inductive step

$$\beta_t(i) = \sum_{j=1}^{N} b_j(z_{t+1}) \beta_{t+1} a_{ij}$$



Solving the smoothing problem

- Use the notation

$$\gamma_t(i) = P(X_t = s_i | z_{1:T})$$

for the probability we want to compute.
- Then

$$\begin{aligned}
\gamma_t(i) &= cP(z_{1:T} | X_t = s_i)P(X_t = s_i) \\
&= cP(z_{1:t} | X_t = s_i)P(z_{t+1:T} | X_t = s_i)P(X_t = s_i) \\
&= cP(z_{1:t} \wedge X_t = s_i)P(z_{t+1:T} | X_t = s_i) \\
&= c\alpha_t(i)\beta_t(i)
\end{aligned}$$

where $c = 1/P(z_{1:T})$ is a constant of proportionality we can ignore as long as we normalize to get the actual probs.

Smoothing

Efficient solution to the *smoothing* problem using the forward and backward variables:

$$P(X_t = s_i | z_{1:T}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

Estimating a state based on all observations before, during, and after that time step.

Forward-backward algorithm

Solving the most probable path problem

- Want $\arg\max_{x_{1:T}} P(x_{1:T} | z_{1:T})$
- One approach:

$$\arg\max_{x_{1:T}} P(x_{1:T} | z_{1:T}) = \arg\max_{x_{1:T}} \frac{P(z_{1:T} | x_{1:T}) P(x_{1:T})}{P(z_{1:T})}$$

$$= \arg\max_{x_{1:T}} P(z_{1:T} | x_{1:T}) P(x_{1:T})$$

- Easy to compute each factor for a given state and observation sequence, but number of paths is exponential in T
- Use dynamic programming instead

DP for Most Probable Path

- Define

$$\delta_t(i) = \max_{x_{1:t-1}} P(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t})$$

- A path giving this maximum is one of length t-1 having the highest probability of simultaneously
  - occuring
  - ending at s_i
  - producing observation sequence $z_{1:t}$

DP for MPP (cont.)

- We'll show that these values can be computed by an efficient forward computation similar to the computation of the α values
- But first, let's check that it gives us something useful:

$$\delta_T(i) = \max_{x_{1:T-1}} P(x_{1:T-1} \wedge X_T = s_i \wedge z_{1:T})$$
$$= \max_{x_{1:T-1}} P(x_{1:T-1} \wedge X_T = s_i | z_{1:T}) P(z_{1:T})$$

- Thus a value of i maximizing $\delta_T(i)$ identifies a state which represents the final state in a path maximizing $P(x_{1:T} | z_{1:T})$

DP for MPP (cont.)

- First, base case is

$$\delta_1(i) = \max_{\text{one choice}} P(X_t = s_i \wedge z_{1:1})$$
$$= P(z_1 | X_t = s_i) P(X_t = s_i)$$
$$= b_i(z_1)\pi_i$$

- Then, since the max. prob. path ending at s_j at time t+1 must go through *some* state at time t, we can write

$$\delta_{t+1}(j) \equiv \max_{x_{1:t}} P(x_{1:t} \wedge X_{t+1} = s_j \wedge z_{1:t+1})$$
$$= \max_i \max_{x_{1:t-1}} \underbrace{P(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t} \wedge z_{t+1} \wedge X_{t+1} = s_j)}$$

Now work on just this part

Call it Δ(i,j)

DP for MPP (cont.)

- Using the chain rule and the Markov property, we find that the probability to be maximized can be written as

$$\Delta(i,j) \equiv P\left(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t} \wedge z_{t+1} \wedge X_{t+1} = s_j\right)$$
$$= P\left(z_{t+1} \wedge X_{t+1} = s_j \middle| x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t}\right) P\left(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t}\right)$$
$$= P\left(z_{t+1} \wedge X_{t+1} = s_j \middle| X_t = s_i\right) P\left(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t}\right)$$
$$= P\left(z_{t+1} \middle| X_{t+1} = s_j\right) P\left(X_{t+1} = s_j \middle| X_t = s_i\right) P\left(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t}\right)$$
$$= b_j(z_{t+1}) a_{ij} P\left(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t}\right)$$

DP for MPP (cont.)

- Finally, then, we get

$$\delta_{t+1}(j) = \max_i \max_{x_{1:t-1}} \Delta(i,j)$$
$$= \max_i \max_{x_{1:t-1}} \left[b_j(z_{t+1}) a_{ij} P\left(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t}\right)\right]$$
$$= b_j(z_{t+1}) \max_i \left[a_{ij} \max_{x_{1:t-1}} P\left(x_{1:t-1} \wedge X_t = s_i \wedge z_{1:t}\right)\right]$$
$$= b_j(z_{t+1}) \max_i a_{ij} \delta_t(i)$$

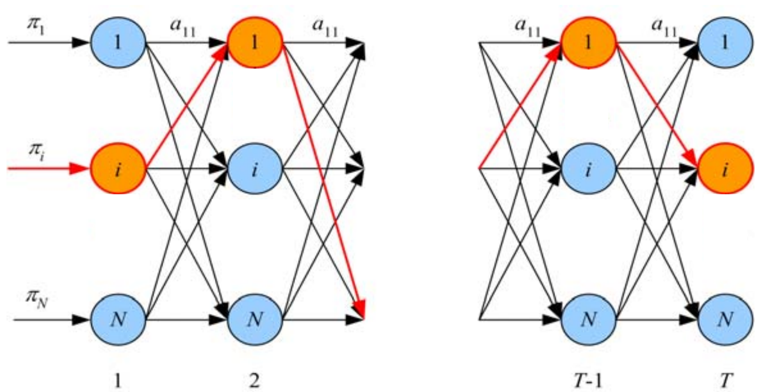
- This is inductive step
- Virtually identical to computation of forward variables α – only difference is that it uses max instead of sum
- Also need to keep track of which state s_i gives max for each state s_j at the next time step to be able to determine actual MPP, not just its probability

Viterbi Algorithm for Most Probable Path

Summary

- Base case: $\forall i \quad \delta_1(i) = b_i(z_1)\pi_i$
- Inductive step: $\forall j \quad \delta_{t+1}(j) = b_j(z_{t+1}) \max_i a_{ij}\delta_t(i)$
- Compute for all states at t=1, then t=2, etc.
- Also save index giving max for each state at each time step (backward pointers)
- Construct the MPP by determining state with largest $\delta_T(i)$, then following backward pointers to time steps T-1, T-2, etc.

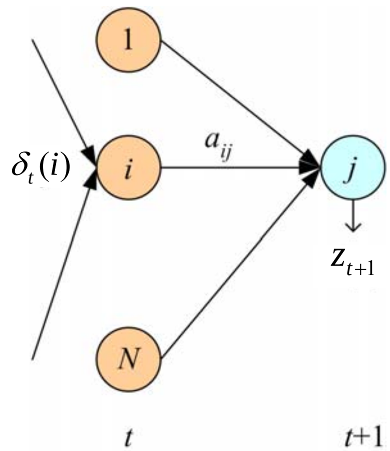
Viterbi Algorithm



Store two numbers at each node in this trellis, one for δ and the other a backward pointer to a node in the previous layer giving the max for this node – this is computed left to right.

To find a most probable path, determine a node in the T layer with max δ value, then follow backward pointers from right to left.

Viterbi algorithm: inductive step



$$\delta_{t+1}(j) = b_j(z_{t+1}) \max_i a_{ij}\delta_t(i)$$
$$\psi_{t+1}(j) = \arg\max_i a_{ij}\delta_t(i)$$

Prob. of a given transition

- The final problem we want to address is the HMM inference (learning) problem, given a training set of observation sequences
- Most of the ingredients for deriving a max. likelihood method for this are in place
- But there's one more sub-problem we'll need to address:

  Given an observation sequence $z_{1:T}$, what's the probability that the state transition s_i to s_j occurred at time t?
- Thus we define

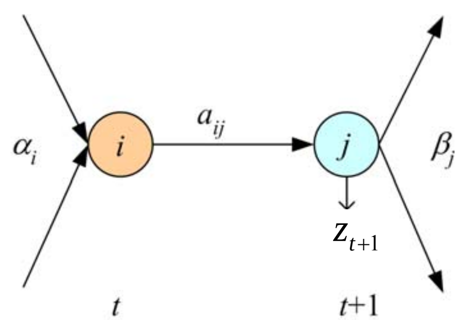
$$\xi_t(i, j) = P\left(X_t = s_i \wedge X_{t+1} = s_j \middle| z_{1:T}\right)$$

Prob. of a given transition (cont.)

$$\begin{aligned}
\xi_t(i,j) &\equiv P\left(X_t = s_i \wedge X_{t+1} = s_j \middle| z_{1:T}\right) \\
&= cP\left(z_{1:T} \middle| X_t = s_i \wedge X_{t+1} = s_j\right)P\left(X_t = s_i \wedge X_{t+1} = s_j\right) \\
&= cP\left(z_{1:T} \middle| X_t = s_i \wedge X_{t+1} = s_j\right)P\left(X_{t+1} = s_j \middle| X_t = s_i\right)P\left(X_t = s_i\right) \\
&= cP\left(z_{1:T} \middle| X_t = s_i \wedge X_{t+1} = s_j\right)a_{ij}P\left(X_t = s_i\right) \\
&= cP\left(z_{1:t} \middle| X_t = s_i\right)P\left(z_{t+1} \middle| X_{t+1} = s_j\right)P\left(z_{t+2:T} \middle| X_{t+1} = s_j\right)a_{ij}P\left(X_t = s_i\right) \\
&= cP\left(z_{1:t} \wedge X_t = s_i\right)b_j\left(z_{t+1}\right)P\left(z_{t+2:T} \middle| X_{t+1} = s_j\right)a_{ij} \\
&= c\alpha_t(i)b_j\left(z_{t+1}\right)\beta_{t+1}(j)a_{ij} \\
&= c\alpha_t(i)a_{ij}b_j\left(z_{t+1}\right)\beta_{t+1}(j)
\end{aligned}$$

c = 1/P($z_{1:T}$) is a normalizing constant we can ignore as long as we make the sum over all (i,j) pairs equal to 1 when computing actual probabilities.

Prob. of a given transition (cont.)



$$\xi_t(i,j) \equiv P\left(X_t = s_i \wedge X_{t+1} = s_j \middle| z_{1:T}\right)$$

$$= \frac{\alpha_t(i)a_{ij}b_j\left(z_{t+1}\right)\beta_{t+1}(j)}{\sum_{k,l}\alpha_t(k)a_{kl}b_l\left(z_{t+1}\right)\beta_{t+1}(l)}$$

Max. Likelihood HMM Inference

Given a state set $\{s_1, s_2, ..., s_N\}$ and a set of R observation sequences

$$z_{1:T_1}^1 = \left(z_1^1, z_2^1, \ldots, z_{T_1}^1\right)$$
$$z_{1:T_2}^2 = \left(z_1^2, z_2^2, \ldots, z_{T_2}^2\right)$$
$$\vdots$$
$$z_{1:T_R}^R = \left(z_1^R, z_2^R, \ldots, z_{T_R}^R\right)$$

determine parameter set $\lambda = (\pi_i, \{a_{ij}\}, \{b_i(o_j)\})$ maximizing

$$\lambda^* = \arg\max_\lambda \prod_{r=1}^{R} P(z_{1:T_r}^r \mid \lambda)$$

From now on, we'll make conditioning on λ explicit

A cheat

Let's first imagine that along with each observation sequence

$$z_{1:T_r}^r = \left(z_1^r, z_2^r, \ldots, z_{T_r}^r\right)$$

an oracle also gives us the corresponding state sequence

$$x_{1:T_r}^r = \left(x_1^r, x_2^r, \ldots, x_{T_r}^r\right)$$

Then we could obtain max. likelihood estimates of all parameters as follows:

$$\hat{\pi}_i = \frac{\#\text{ of sequences starting with } s_i}{\text{total }\#\text{ of sequences}}$$

$$\hat{a}_{ij} = \frac{\#\text{ of transitions } s_i \to s_j}{\#\text{ of visits to state } s_i}$$

$$\hat{b}_i(o_k) = \frac{\#\text{ of visits to state } s_i \text{ where } o_k \text{ observed}}{\#\text{ visits to state } s_i}$$

A cheat (cont.)

More formally, define the indicator functions

$$\chi_t^r(i) = \begin{cases} 1 & \text{if } x_t^r = s_i \\ 0 & \text{otherwise} \end{cases}$$

$$\chi_t^r(i \to j) = \begin{cases} 1 & \text{if } x_t^r = s_i \text{ and } x_{t+1}^r = s_j \\ 0 & \text{otherwise} \end{cases}$$

$$\chi_t^r(i : k) = \begin{cases} 1 & \text{if } x_t^r = s_i \text{ and } z_t^r = o_k \\ 0 & \text{otherwise} \end{cases}$$

Hidden Markov Models: Slide 83

A cheat (cont.)

In terms of these indicator functions, our ML estimates would then be

$$\hat{\pi}_i = \frac{\sum_{r=1}^{R} \chi_1^r(i)}{R}$$

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r - 1} \chi_t^r(i \to j)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r - 1} \chi_t^r(i)}$$

For this, we can't use the last state in any of the training sequences because there's no next state

$$\hat{b}_i(o_k) = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \chi_t^r(i : k)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \chi_t^r(i)}$$

Hidden Markov Models: Slide 84

The bad news ...

- There is no oracle to tell us the state sequence corresponding to each observation sequence
- So we don't know these actual indicator function values
- So we can't compute these sums

The good news ...

- We can compute their expected values efficiently:

$$\gamma_t^r(i) \equiv P\left(X_t = s_i \middle| z_{1:T_r}^r, \lambda\right) = E\left(\chi_t^r(i) \middle| \lambda\right)$$

$$\xi_t^r(i,j) \equiv P\left(X_t = s_i \wedge X_{t+1} = s_j \middle| z_{1:T_r}^r, \lambda\right) = E\left(\chi_t^r(i \to j) \middle| \lambda\right)$$

- Also:

$$E\left(\chi_t^r(i:k) \middle| \lambda\right) = P\left(X_t = s_i \wedge Z_t = o_k \middle| z_{1:T}^r, \lambda\right)$$

$$= \begin{cases} P\left(X_t = s_i \middle| z_{1:T}^r, \lambda\right) & \text{if } z_t^r = o_k \\ 0 & \text{otherwise} \end{cases}$$

$$= \gamma_t^r(i) \underbrace{I\left(z_t^r = o_k\right)}_{\substack{\text{Usual indicator function:} \\ \text{1 if true, 0 if false}}}$$

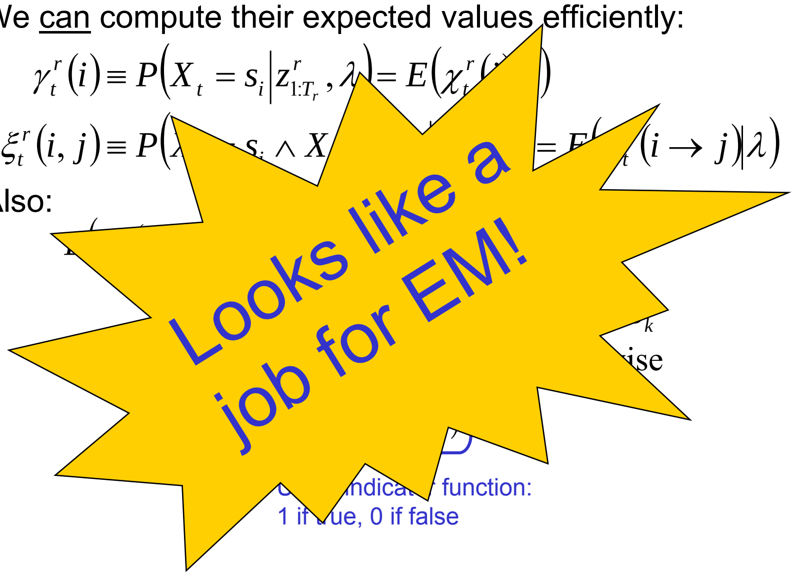
The good news ...

- We can compute their expected values efficiently:

$$\gamma_t^r(i) \equiv P\left(X_t = s_i \middle| z^r_{1:T_r}, \lambda\right) = E\left(\chi_t^r(i) \ldots\right)$$

$$\xi_t^r(i,j) \equiv P\left(X \ldots = s_i \wedge X \ldots\right) = E\left(\ldots_t(i \to j) \middle| \lambda\right)$$

- Also:



...ise

...indica... function:
1 if ...ue, 0 if false

EM for HMMs (Baum-Welch)

E-step

Use the current estimate of model parameters λ to compute all the $\gamma_t^r(i)$ and $\xi_t^r(i,j)$ values for each training sequence $z^r_{1:T_r}$.

M-step

$$\pi_i \leftarrow \frac{\sum_{r=1}^{R} \gamma_1^r(i)}{R} \qquad a_{ij} \leftarrow \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r - 1} \xi_t^r(i,j)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r - 1} \gamma_t^r(i)}$$

$$b_i(k) \leftarrow \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_t^r(i) I\left(z_t^r = o_k\right)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_t^r(i)}$$

Remarks on Baum-Welch

- Bad news: There may be many local maxima
- Good news: The local maxima are usually adequate models of the data
- Any probabilities initialized to zero will remain zero throughout – useful when one wants a model with limited state transitions

Summary of solution methods

- Filtering: forward variables (α's)
- Prediction: (modified) forward variables
- Smoothing: forward-backward algorithm
- Observation sequence likelihood: forward variables
- Most probable path: Viterbi algorithm
- Maximum likelihood model: Baum-Welch algorithm

Some good references

- Standard HMM reference:

  L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. of the IEEE, Vol.77, No.2, pp.257-286, 1989.

- Excellent reference for Dynamic Bayes Nets as a unifying framework for probabilistic temporal models (including HMMs and Kalman filters):

  Chapter 15 of *Artificial Intelligence, A Modern Approach, 2nd Edition*, by Russell & Norvig

What You Should Know

- What an HMM is
- Definition, computation, and use of $\alpha_t(i)$
- The Viterbi algorithm
- Outline of the EM algorithm for HMM learning (Baum-Welch)
- Be comfortable with the kind of math needed to derive the HMM algorithms described here
- What a DBN is and how an HMM is a special case
- Appreciate that a DBN (and thus an HMM) is really just a special kind of Bayes net