Notes

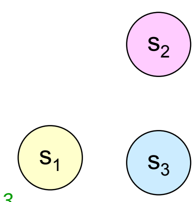
- Class Schedule
- Simon Baker RI Seminar Friday 3:30 Face Tracking NSH 1305

Hidden Markov Models

Andrew W. Moore
Professor
School of Computer Science
Carnegie Mellon University
www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

 Nov 29th, 2001

A Markov System



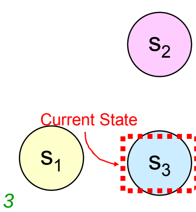
Has N states, called $s_1, s_2 \, .. \, s_N$

There are discrete timesteps, $t=0, t=1, \ldots$

$N = 3$

$t=0$

A Markov System



Has N states, called $s_1, s_2 \, .. \, s_N$

There are discrete timesteps, $t=0, t=1, \ldots$

On the t'th timestep the system is in exactly one of the available states. Call it q_t

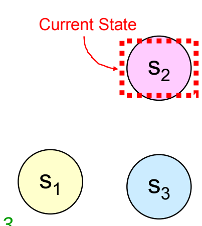
Note: $q_t \in \{s_1, s_2 \, .. \, s_N\}$

$N = 3$

$t=0$

$q_t=q_0=s_3$

A Markov System



Has N states, called $s_1, s_2 \, .. \, s_N$

There are discrete timesteps, $t=0, t=1, \ldots$

On the t'th timestep the system is in exactly one of the available states. Call it q_t

Note: $q_t \in \{s_1, s_2 \, .. \, s_N\}$

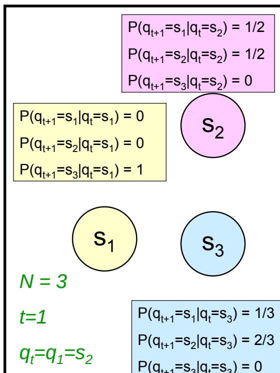
Between each timestep, the next state is chosen randomly.

$N = 3$

$t=1$

$q_t=q_1=s_2$

A Markov System



Has N states, called $s_1, s_2 \, .. \, s_N$

There are discrete timesteps, $t=0, t=1, \ldots$

On the t'th timestep the system is in exactly one of the available states. Call it q_t

Note: $q_t \in \{s_1, s_2 \, .. \, s_N\}$

Between each timestep, the next state is chosen randomly.

The current state determines the probability distribution for the next state.

$N = 3$

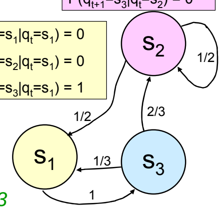
$t=1$

$q_t=q_1=s_2$

A Markov System

$P(q_{t+1}=s_1|q_t=s_2) = 1/2$
$P(q_{t+1}=s_2|q_t=s_2) = 1/2$
$P(q_{t+1}=s_3|q_t=s_2) = 0$

$P(q_{t+1}=s_1|q_t=s_1) = 0$
$P(q_{t+1}=s_2|q_t=s_1) = 0$
$P(q_{t+1}=s_3|q_t=s_1) = 1$



$P(q_{t+1}=s_1|q_t=s_3) = 1/3$
$P(q_{t+1}=s_2|q_t=s_3) = 2/3$
$P(q_{t+1}=s_3|q_t=s_3) = 0$

N = 3

t=1

$q_t=q_1=s_2$

Often notated with arcs between states

Has *N* states, called $s_1, s_2 .. s_N$

There are discrete timesteps, *t=0, t=1, …*

On the t'th timestep the system is in exactly one of the available states. Call it q_t

Note: $q_t \in \{s_1, s_2 .. s_N\}$

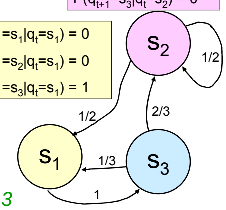
Between each timestep, the next state is chosen randomly.

The current state determines the probability distribution for the next state.

Markov Property

$P(q_{t+1}=s_1|q_t=s_2) = 1/2$
$P(q_{t+1}=s_2|q_t=s_2) = 1/2$
$P(q_{t+1}=s_3|q_t=s_2) = 0$

$P(q_{t+1}=s_1|q_t=s_1) = 0$
$P(q_{t+1}=s_2|q_t=s_1) = 0$
$P(q_{t+1}=s_3|q_t=s_1) = 1$



$P(q_{t+1}=s_1|q_t=s_3) = 1/3$
$P(q_{t+1}=s_2|q_t=s_3) = 2/3$
$P(q_{t+1}=s_3|q_t=s_3) = 0$

N = 3

t=1

$q_t=q_1=s_2$

q_{t+1} is conditionally independent of $\{ q_{t-1}, q_{t-2}, \ldots q_1, q_0 \}$ given q_t.

In other words:

$P(q_{t+1} = s_j | q_t = s_i) =$

$P(q_{t+1} = s_j | q_t = s_i$,any earlier history$)$

Question: what would be the best Bayes Net structure to represent the Joint Distribution of ($q_0, q_1, \ldots q_3, q_4$)?

Hidden Markov Models

- Question 1: State Estimation
  What is $P(q_T=S_i \mid O_1O_2\ldots O_T)$
  It will turn out that a new cute D.P. trick will get this for us.
- Question 2: Most Probable Path
  Given $O_1O_2\ldots O_T$, what is the most probable path that I took?
  And what is that probability?
  Yet another famous D.P. trick, the VITERBI algorithm, gets this.
- Question 3: Learning HMMs:
  Given $O_1O_2\ldots O_T$, what is the maximum likelihood HMM that could have produced this string of observations?
  Very very useful. Uses the E.M. Algorithm

Are H.M.M.s Useful?

You bet !!

- Robot planning + sensing when there's uncertainty
- Speech Recognition/Understanding
  Phones → Words, Signal → 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.

HMM Notation (from Rabiner's Survey)

*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.

The states are labeled $S_1\ S_2\ ..\ S_N$

For a particular trial….

Let T be the number of observations

T is also the number of states passed through

$O = O_1\ O_2\ ..\ O_T$ is the sequence of observations

$Q = q_1\ q_2\ ..\ q_T$ is the notation for a path of states

$\lambda = \langle N,M,\{\pi_i\},\{a_{ij}\},\{b_i(j)\}\rangle$ is the specification of an HMM

HMM Formal Definition

An HMM, λ, is a 5-tuple consisting of

- N the number of states
- M the number of possible observations
- $\{\pi_1, \pi_2, .. \pi_N\}$ The starting state probabilities
  $P(q_0 = S_i) = \pi_i$

This is new. In our previous example, start state was deterministic

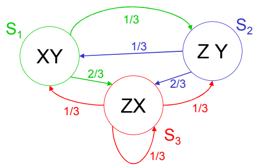
- The state transition probabilities $P(q_{t+1}=S_j \mid q_t=S_i)=a_{ij}$

$$\begin{matrix} a_{11} & a_{22} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ : & : & & : \\ a_{N1} & a_{N2} & \ldots & a_{NN} \end{matrix}$$

- The observation probabilities $P(O_t=k \mid q_t=S_i)=b_i(k)$

$$\begin{matrix} b_1(1) & b_1(2) & \ldots & b_1(M) \\ b_2(1) & b_2(2) & \ldots & b_2(M) \\ : & : & & : \\ b_N(1) & b_N(2) & \ldots & b_N(M) \end{matrix}$$

Here's an HMM



Start randomly in state 1 or 2

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

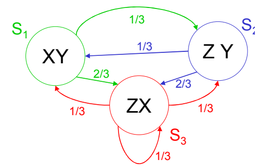
N = 3
M = 3

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 (X) = 1/2 b_1 (Y) = 1/2 b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = 1/2 b_2 (Z) = 1/2
b_3 (X) = 1/2 b_3 (Y) = 0 b_3 (Z) = 1/2

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

q_0=	___	O_0=	___
q_1=	___	O_1=	___
q_2=	___	O_2=	___

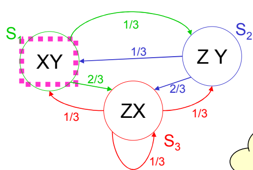
N = 3
M = 3

π_1 = ½ π_2 = ½ π_3 = 0

a_{11} = 0 a_{12} = ⅓ a_{13} = ⅔
a_{12} = ⅓ a_{22} = 0 a_{13} = ⅔
a_{13} = ⅓ a_{32} = ⅓ a_{13} = ⅓

b_1 (X) = ½ b_1 (Y) = ½ b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = ½ b_2 (Z) = ½
b_3 (X) = ½ b_3 (Y) = 0 b_3 (Z) = ½

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 X and Y

q_0=	S_1	O_0=	___
q_1=	___	O_1=	___
q_2=	___	O_2=	___

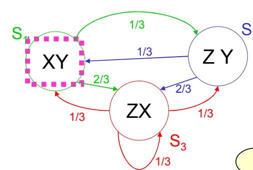
N = 3
M = 3

π_1 = ½ π_2 = ½ π_3 = 0

a_{11} = 0 a_{12} = ⅓ a_{13} = ⅔
a_{12} = ⅓ a_{22} = 0 a_{13} = ⅔
a_{13} = ⅓ a_{32} = ⅓ a_{13} = ⅓

b_1 (X) = ½ b_1 (Y) = ½ b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = ½ b_2 (Z) = ½
b_3 (X) = ½ b_3 (Y) = 0 b_3 (Z) = ½

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:

Goto S_3 with probability 2/3 or S_2 with prob. 1/3

q_0=	S_1	O_0=	X
q_1=	___	O_1=	___
q_2=	___	O_2=	___

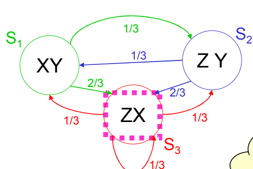
N = 3
M = 3

π_1 = ½ π_2 = ½ π_3 = 0

a_{11} = 0 a_{12} = ⅓ a_{13} = ⅔
a_{12} = ⅓ a_{22} = 0 a_{13} = ⅔
a_{13} = ⅓ a_{32} = ⅓ a_{13} = ⅓

b_1 (X) = ½ b_1 (Y) = ½ b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = ½ b_2 (Z) = ½
b_3 (X) = ½ b_3 (Y) = 0 b_3 (Z) = ½

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 Z and X

q_0=	S_1	O_0=	X
q_1=	S_3	O_1=	___
q_2=	___	O_2=	___

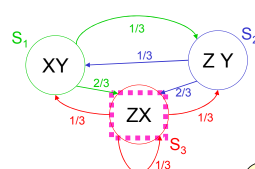
N = 3
M = 3

π_1 = ½ π_2 = ½ π_3 = 0

a_{11} = 0 a_{12} = ⅓ a_{13} = ⅔
a_{12} = ⅓ a_{22} = 0 a_{13} = ⅔
a_{13} = ⅓ a_{32} = ⅓ a_{13} = ⅓

b_1 (X) = ½ b_1 (Y) = ½ b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = ½ b_2 (Z) = ½
b_3 (X) = ½ b_3 (Y) = 0 b_3 (Z) = ½

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:

Each of the three next states is equally likely

q_0=	S_1	O_0=	X
q_1=	S_3	O_1=	X
q_2=	___	O_2=	___

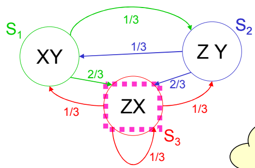
N = 3
M = 3

π_1 = ½ π_2 = ½ π_3 = 0

a_{11} = 0 a_{12} = ⅓ a_{13} = ⅔
a_{12} = ⅓ a_{22} = 0 a_{13} = ⅔
a_{13} = ⅓ a_{32} = ⅓ a_{13} = ⅓

b_1 (X) = ½ b_1 (Y) = ½ b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = ½ b_2 (Z) = ½
b_3 (X) = ½ b_3 (Y) = 0 b_3 (Z) = ½

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 Z and X

N = 3
M = 3

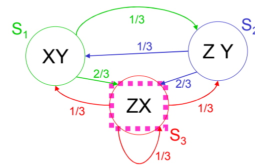
$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$	$a_{12} = ⅓$	$a_{13} = ⅔$
$a_{12} = ⅓$	$a_{22} = 0$	$a_{13} = ⅔$
$a_{13} = ⅓$	$a_{32} = ⅓$	$a_{13} = ⅓$

$b_1 (X) = ½$	$b_1 (Y) = ½$	$b_1 (Z) = 0$
$b_2 (X) = 0$	$b_2 (Y) = ½$	$b_2 (Z) = ½$
$b_3 (X) = ½$	$b_3 (Y) = 0$	$b_3 (Z) = ½$

q_0=	S$_1$	O_0=	X
q_1=	S$_3$	O_1=	X
q_2=	S$_3$	O_2=	__

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:

N = 3
M = 3

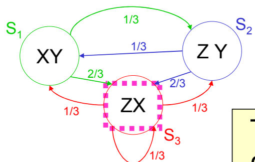
$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$	$a_{12} = ⅓$	$a_{13} = ⅔$
$a_{12} = ⅓$	$a_{22} = 0$	$a_{13} = ⅔$
$a_{13} = ⅓$	$a_{32} = ⅓$	$a_{13} = ⅓$

$b_1 (X) = ½$	$b_1 (Y) = ½$	$b_1 (Z) = 0$
$b_2 (X) = 0$	$b_2 (Y) = ½$	$b_2 (Z) = ½$
$b_3 (X) = ½$	$b_3 (Y) = 0$	$b_3 (Z) = ½$

q_0=	S$_1$	O_0=	X
q_1=	S$_3$	O_1=	X
q_2=	S$_3$	O_2=	Z

State Estimation



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…

N = 3
M = 3

$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$	$a_{12} = ⅓$	$a_{13} = ⅔$
$a_{12} = ⅓$	$a_{22} = 0$	$a_{13} = ⅔$
$a_{13} = ⅓$	$a_{32} = ⅓$	$a_{13} = ⅓$

$b_1 (X) = ½$	$b_1 (Y) = ½$	$b_1 (Z) = 0$
$b_2 (X) = 0$	$b_2 (Y) = ½$	$b_2 (Z) = ½$
$b_3 (X) = ½$	$b_3 (Y) = 0$	$b_3 (Z) = ½$

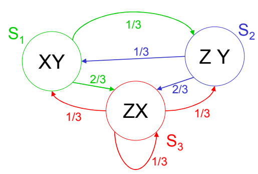
q_0=	?	O_0=	X
q_1=	?	O_1=	X
q_2=	?	O_2=	Z

Bayes' Rule

- $\mathbf{O}=\{O_1 O_2 \ldots O_T\}$
- $P(Q_T|\mathbf{O}) = P(\mathbf{O}|Q_T)\, P(\mathbf{O}) / P(Q_T)$

Prob. of a series of observations

What is $P(\mathbf{O}) = P(O_1\ O_2\ O_3) = P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?



Slow, stupid way:

$$P(\mathbf{O}) = \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \wedge \mathbf{Q})$$

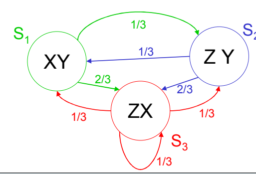
$$= \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \mid \mathbf{Q}) P(\mathbf{Q})$$

How do we compute P(Q) for an arbitrary path Q?

How do we compute P(O|Q) for an arbitrary path Q?

Prob. of a series of observations

What is $P(\mathbf{O}) = P(O_1\ O_2\ O_3) = P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?



Slow, stupid way:

$$P(\mathbf{O}) = \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \wedge \mathbf{Q})$$

$$= \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \mid \mathbf{Q}) P(\mathbf{Q})$$

How do we compute P(Q) for an arbitrary path Q?

$P(Q)= P(q_1,q_2,q_3)$

$=P(q_1)\ P(q_2,q_3|q_1)$ (chain rule)

$=P(q_1)\ P(q_2|q_1)\ P(q_3|\ q_2,q_1)$ (chain)

$=P(q_1)\ P(q_2|q_1)\ P(q_3|\ q_2)$ (why?)

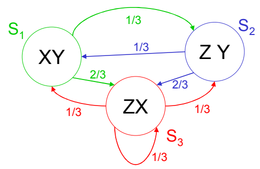
Example in the case $Q = S_1 S_3 S_3$:

=1/2 * 2/3 * 1/3 = 1/9

How do we compute P(O|Q) for an arbitrary path Q?

Prob. of a series of observations

What is P(**O**) = $P(O_1\, O_2\, O_3)$ = $P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?



Slow, stupid way:

$$P(\mathbf{O}) = \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \wedge \mathbf{Q})$$

$$= \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \mid \mathbf{Q}) P(\mathbf{Q})$$

How do we compute P(Q) for an arbitrary path Q?

How do we compute P(O|Q) for an arbitrary path Q?

P(O|Q)

= $P(O_1\, O_2\, O_3 \mid q_1\, q_2\, q_3)$

= $P(O_1 \mid q_1)\, P(O_2 \mid q_2)\, P(O_3 \mid q_3)$ (why?)

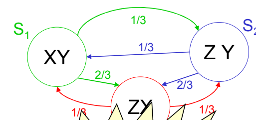
Example in the case Q = $S_1\, S_3\, S_3$:

= $P(X \mid S_1)\, P(X \mid S_3)\, P(Z \mid S_3)$ =

=1/2 * 1/2 * 1/2 = 1/8

Prob. of a series of observations

What is P(**O**) = $P(O_1\, O_2\, O_3)$ = $P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?



Slow, stupid way:

$$P(\mathbf{O}) = \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \wedge \mathbf{Q})$$

$$= \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \mid \mathbf{Q}) P(\mathbf{Q})$$

How do we compute P(Q) for an arbitrary path Q?

How do we compute P(O|Q) for an arbitrary path Q?

P(**O**) would need 27 P(Q) computations and 27 P(O|Q) computations

A sequence of 20 observations would need 3^{20} = 3.5 billion computations and 3.5 billion P(O|Q) computations

So let's be smarter…

The Prob. of a given series of observations, non-exponential-cost-style

Given observations $O_1\, O_2 \ldots O_T$

Define

$\alpha_t(i) = P(O_1\, O_2 \ldots O_t \wedge q_t = S_i \mid \lambda)$ where $1 \le t \le T$

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

- We'd have seen the first t observations
- We'd have ended up in S_i as the t'th state visited.

In our example, what is $\alpha_2(3)$?

$\alpha_t(i)$: easy to define recursively

$\alpha_t(i) = P(O_1\, O_2 \ldots O_T \wedge q_t = S_i \mid \lambda)$ ($\alpha_t(i)$ can be defined stupidly by considering all paths length "t". How?)

$$\alpha_1(i) = P(O_1 \wedge q_1 = S_i)$$
$$= P(q_1 = S_i) P(O_1 \mid q_1 = S_i)$$
$$= \qquad \text{what?}$$
$$\alpha_{t+1}(j) = P(O_1 O_2 \ldots O_t O_{t+1} \wedge q_{t+1} = S_j)$$
$$=$$

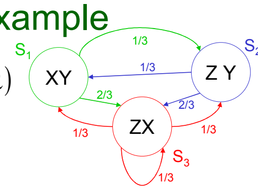
$\alpha_t(i)$: easy to define recursively

$\alpha_t(i) = P(O_1\, O_2 \ldots O_T \wedge q_t = S_i \mid \lambda)$ ($\alpha_t(i)$ can be defined stupidly by considering all paths length "t". How?)

$$\alpha_1(i) = P(O_1 \wedge q_1 = S_i)$$
$$= P(q_1 = S_i) P(O_1 \mid q_1 = S_i)$$
$$= \qquad \text{what?}$$
$$\alpha_{t+1}(j) = P(O_1 O_2 \ldots O_t O_{t+1} \wedge q_{t+1} = S_j)$$
$$= \sum_{i=1}^{N} P(O_1 O_2 \ldots O_t \wedge q_t = S_i \wedge O_{t+1} \wedge q_{t+1} = S_j)$$
$$= \sum_{i=1}^{N} P(O_{t+1}, q_{t+1} = S_j \mid O_1 O_2 \ldots O_t \wedge q_t = S_i) P(O_1 O_2 \ldots O_t \wedge q_t = S_i)$$
$$= \sum_i P(O_{t+1}, q_{t+1} = S_j \mid q_t = S_i) \alpha_t(i)$$
$$= \sum_i P(q_{t+1} = S_j \mid q_t = S_i) P(O_{t+1} \mid q_{t+1} = S_j) \alpha_t(i)$$
$$= \sum_i a_{ij} b_j(O_{t+1}) \alpha_t(i)$$

in our example

$$\alpha_t(i) = P(O_1 O_2 .. O_t \wedge q_t = S_i \mid \lambda)$$
$$\alpha_1(i) = b_i(O_1) \pi_i$$
$$\alpha_{t+1}(j) = \sum_i a_{ij} b_j(O_{t+1}) \alpha_t(i)$$



WE SAW $O_1\, O_2\, O_3$ = X X Z

$\alpha_1(1) = \frac{1}{4}$ $\quad \alpha_1(2) = 0$ $\quad \alpha_1(3) = 0$

$\alpha_2(1) = 0$ $\quad \alpha_2(2) = 0$ $\quad \alpha_2(3) = \frac{1}{12}$

$\alpha_3(1) = 0$ $\quad \alpha_3(2) = \frac{1}{72}$ $\quad \alpha_3(3) = \frac{1}{72}$

Easy Question

We can cheaply compute

$\alpha_t(i) = P(O_1 O_2 \ldots O_t \wedge q_t = S_i)$

(How) can we cheaply compute

$P(O_1 O_2 \ldots O_t)$?

(How) can we cheaply compute

$P(q_t = S_i | O_1 O_2 \ldots O_t)$

Easy Question

We can cheaply compute

$\alpha_t(i) = P(O_1 O_2 \ldots O_t \wedge q_t = S_i)$

(How) can we cheaply compute

$P(O_1 O_2 \ldots O_t)$? $\sum_{i=1}^{N} \alpha_t(i)$

(How) can we cheaply compute

$P(q_t = S_i | O_1 O_2 \ldots O_t)$ $\frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}$

Most probable path given observations

What's most probable path given $O_1 O_2 \ldots O_T$, i.e.

What is $\underset{Q}{\operatorname{argmax}} \, P(Q | O_1 O_2 \ldots O_T)$?

Slow, stupid answer:

$$\underset{Q}{\operatorname{argmax}} \, P(Q | O_1 O_2 \ldots O_T)$$

$$= \underset{Q}{\operatorname{argmax}} \frac{P(O_1 O_2 \ldots O_T | Q) P(Q)}{P(O_1 O_2 \ldots O_T)}$$

$$= \underset{Q}{\operatorname{argmax}} \, P(O_1 O_2 \ldots O_T | Q) P(Q)$$

Efficient MPP computation

We're going to compute the following variables:

$\delta_t(i) = \max_{q_1 q_2 .. q_{t-1}} P(q_1 q_2 .. q_{t-1} \wedge q_t = S_i \wedge O_1 .. O_t)$

= The Probability of the path of Length t-1 with the maximum chance of doing all these things:

...OCCURING

and

...ENDING UP IN STATE S_i

and

...PRODUCING OUTPUT $O_1 \ldots O_t$

DEFINE: $mpp_t(i)$ = that path

So: $\delta_t(i)$ = Prob($mpp_t(i)$)

The Viterbi Algorithm

$$\delta_t(i) = \max_{q_1 q_2 \ldots q_{t-1}} P(q_1 q_2 \ldots q_{t-1} \wedge q_t = S_i \wedge O_1 O_2 .. O_t)$$

$$mpp_t(i) = \underset{q_1 q_2 \ldots q_{t-1}}{\operatorname{argmax}} P(q_1 q_2 \ldots q_{t-1} \wedge q_t = S_i \wedge O_1 O_2 .. O_t)$$

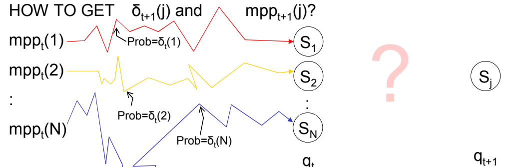
$$\delta_1(i) = \max_{\text{one choice}} P(q_1 = S_i \wedge O_1)$$

$$= P(q_1 = S_i) P(O_1 | q_1 = S_i)$$

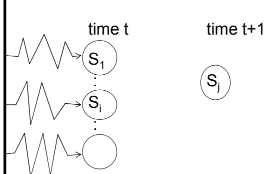
$$= \pi_i b_i(O_1)$$

Now, suppose we have all the $\delta_t(i)$'s and $mpp_t(i)$'s for all i.

HOW TO GET $\delta_{t+1}(j)$ and $mpp_{t+1}(j)$?



The Viterbi Algorithm

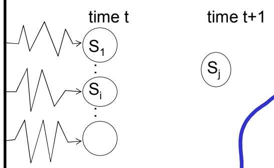


The most prob path with last two states S_i S_j

is

the most prob path to S_i, followed by transition $S_i \rightarrow S_j$

The Viterbi Algorithm



The most prob path with last two states S_i S_j

is

the most prob path to S_i , followed by transition $S_i \rightarrow S_j$

What is the prob of that path?

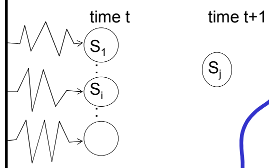
$\delta_t(i) \times P(S_i \rightarrow S_j \wedge O_{t+1} \mid \lambda)$

$= \delta_t(i)\, a_{ij}\, b_j(O_{t+1})$

SO The most probable path to S_j has S_{i^*} as its penultimate state

where $i^* = \underset{i}{\operatorname{argmax}}\, \delta_t(i)\, a_{ij}\, b_j(O_{t+1})$

The Viterbi Algorithm



The most prob path with last two states S_i S_j

is

the most prob path to S_i , followed by transition $S_i \rightarrow S_j$

What is the prob of that path?

$\delta_t(i) \times P(S_i \rightarrow S_j \wedge O_{t+1} \mid \lambda)$

$= \delta_t(i)\, a_{ij}\, b_j(O_{t+1})$

SO The most probable path to S_j has S_{i^*} as its penultimate state

where $i^* = \underset{i}{\operatorname{argmax}}\, \delta_t(i)\, a_{ij}\, b_j(O_{t+1})$

Summary:

$\delta_{t+1}(j) = \delta_t(i^*)\, a_{ij}\, b_j(O_{t+1})$
$mpp_{t+1}(j) = mpp_{t+1}(i^*) S_{i^*}$
} with i* defined to the left

What's Viterbi used for?

Classic Example

Speech recognition:

Signal $\rightarrow$ words

HMM $\rightarrow$ observable is signal

$\rightarrow$ Hidden state is part of word formation

What is the most probable word given this signal?

UTTERLY GROSS SIMPLIFICATION

In practice: many levels of inference; not one big jump.

HMMs are used and useful

But how do you design an HMM?

Occasionally, (e.g. in our robot example) it is reasonable to deduce the HMM from first principles.

But usually, especially in Speech or Genetics, it is better to infer it from large amounts of data. $O_1\ O_2\ ..\ O_T$ with a big "T".

Observations previously in lecture ⇒ $O_1\ O_2\ ..\ O_T$

Observations in the next bit ⇒ $O_1\ O_2\ ..\ O_{\mathbf{T}}$

Inferring an HMM

Remember, we've been doing things like

$P(O_1\ O_2\ ..\ O_T \mid \lambda)$

That "λ" is the notation for our HMM parameters.

<u>Now</u> We have some observations and we want to estimate λ from them.

AS USUAL: We could use

(i) MAX LIKELIHOOD $\lambda = \underset{\lambda}{\operatorname{argmax}}\, P(O_1\ ..\ O_T \mid \lambda)$

(ii) BAYES

Work out $P(\lambda \mid O_1\ ..\ O_T)$

and then take $E[\lambda]$ or $\underset{\lambda}{\max}\, P(\lambda \mid O_1\ ..\ O_T)$

Max likelihood HMM estimation

Define

$\gamma_t(i) = P(q_t = S_i \mid O_1 O_2 \ldots O_T, \lambda)$

$\varepsilon_t(i,j) = P(q_t = S_i \wedge q_{t+1} = S_j \mid O_1 O_2 \ldots O_T, \lambda)$

$\gamma_t(i)$ and $\varepsilon_t(i,j)$ can be computed efficiently $\forall i,j,t$

(Details in Rabiner paper)

$\sum_{t=1}^{T-1} \gamma_t(i) =$ Expected number of transitions out of state i during the path

$\sum_{t=1}^{T-1} \varepsilon_t(i,j) =$ Expected number of transitions from state i to state j during the path

HMM estimation

$$\gamma_t(i) = P(q_t = S_i \mid O_1 O_2 .. O_T, \lambda)$$

$$\varepsilon_t(i,j) = P(q_t = S_i \wedge q_{t+1} = S_j \mid O_1 O_2 .. O_T, \lambda)$$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions out of state i during path}$$

$$\sum_{t=1}^{T-1} \varepsilon_t(i,j) = \text{expected number of transitions out of i and into j during path}$$

$$\text{Notice } \frac{\sum_{t=1}^{T-1} \varepsilon_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\begin{pmatrix}\text{expected frequency} \\ i \to j\end{pmatrix}}{\begin{pmatrix}\text{expected frequency} \\ i\end{pmatrix}}$$

$$= \text{Estimate of Prob}(\text{Next state } S_j \mid \text{This state } S_i)$$

We can re-estimate

$$a_{ij} \leftarrow \frac{\sum \varepsilon_t(i,j)}{\sum \gamma_t(i)}$$

We can also re-estimate

$$b_j(O_k) \leftarrow \cdots \qquad \text{(See Rabiner)}$$

EM for HMMs

If we knew λ we could estimate EXPECTATIONS of quantities such as
- Expected number of times in state i
- Expected number of transitions i → j

If we knew the quantities such as
- Expected number of times in state i
- Expected number of transitions i → j

We could compute the MAX LIKELIHOOD estimate of

$$\lambda = \langle \{a_{ij}\}, \{b_i(j)\}, \pi_i \rangle$$

Roll on the EM Algorithm…

EM 4 HMMs

1. Get your observations $O_1 \ldots O_T$
2. Guess your first λ estimate λ(0), k=0
3. k = k+1
4. Given $O_1 \ldots O_T$, λ(k) compute
   $\gamma_t(i)$, $\varepsilon_t(i,j)$ $\quad \forall 1 \le t \le T, \quad \forall 1 \le i \le N, \quad \forall 1 \le j \le N$
5. Compute expected freq. of state i, and expected freq. i→j
6. Compute new estimates of a_{ij}, $b_j(k)$, π_i accordingly. Call them λ(k+1)
7. Goto 3, unless converged.

- **Also known (for the HMM case) as the BAUM-WELCH algorithm.**

Bad News

- There are lots of local minima

Good News

- The local minima are usually adequate models of the data.

Notice

- EM does not estimate the number of states. That must be given.
- Often, HMMs are forced to have some links with zero probability. This is done by setting $a_{ij}=0$ in initial estimate λ(0)
- Easy extension of everything seen today: HMMs with real valued outputs

Bad News

- There are lots o
- The local minim data.

Trade-off between too few states (inadequately modeling the structure in the data) and too many (fitting the noise).

Thus #states is a regularization parameter.

Blah blah blah… bias variance tradeoff…blah blah…cross-validation…blah blah….AIC, BIC….blah blah (same ol' same ol')

Notice

- EM does not estimate the number of states. That must be given.
- Often, HMMs are forced to have some links with zero probability. This is done by setting $a_{ij}=0$ in initial estimate λ(0)
- Easy extension of everything seen today: HMMs with real valued outputs

What You Should Know

- What is an HMM ?
- Computing (and defining) $\alpha_t(i)$
- The Viterbi algorithm
- Outline of the EM algorithm
- To be very happy with the kind of maths and analysis needed for HMMs
- Fairly thorough reading of Rabiner* up to page 266* [Up to but not including "IV. Types of HMMs"].

DON'T PANIC: starts on p. 257.

*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.