A Markov System

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

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: $q_t \in \{s_1, s_2 \ldots s_N\}$

Between each timestep, the next state is chosen randomly.
A Markov System

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

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: $q_t \in \{s_1, s_2 \ldots s_N\}$

Between each timestep, the next state is chosen randomly.

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

Markov Property

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

In other words:

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

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

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

Answer:

\[
P(q_{t+1} = s_1 | q_t = s_2) = \frac{1}{2}
\]

$P(q_{t+1} = s_2 | q_t = s_2) = \frac{1}{2}$

Each of these probability tables is identical:

The Joint Distribution of $(q_0, q_1, q_2, q_3, q_4)$?

Notation:

\[
a_{ij} = P(q_{t+1} = s_j | q_t = s_i)
\]

**Dynamics of System**

$q_0 = \begin{array}{c}
    \text{R} \\
    \text{H}
\end{array}$

Each timestep the human moves randomly to an adjacent cell. And Robot also moves randomly to an adjacent cell.

Typical Questions:

- “What’s the expected time until the human is crushed like a bug?”
- “What’s the probability that the robot will hit the left wall before it hits the human?”
- “What’s the probability Robot crushes human on next time step?”

**Example Question**

“It’s currently time $t$, and human remains uncrushed. What’s the probability of crushing occurring at time $t + 1$?”

If robot is blind:

We can compute this in advance.

If robot is omnipotent:

(I.E. If robot knows state at time $t$), can compute directly.

If robot has some sensors, but incomplete state information ...

Hidden Markov Models are applicable!
What is $P(q_t = s)$? slow, stupid answer

Step 1: Work out how to compute $P(Q)$ for any path $Q$ = $q_1 \ q_2 \ q_3 \ldots \ q_t$

Given we know the start state $q_1$ (i.e. $P(q_1) = 1$)

\[ P(q_1\ q_2\ \ldots\ q_t) = P(q_1\ q_2\ \ldots\ q_{t-1})\ P(q_t|q_1\ q_2\ \ldots\ q_{t-1}) \]

WHY?

\[ = P(q_2|q_1)P(q_3|q_2)\ldots P(q_t|q_{t-1}) \]

Step 2: Use this knowledge to get $P(q_t = s)$

\[ P(q_t = s) = \sum_{Q \in \text{Paths of length } t \text{ that end in } s} P(Q) \]

Computation is exponential in $t$

What is $P(q_t = s)$? Clever answer

• For each state $s_i$, define

\[ p_i(i) = \text{Prob. state is } s_i \text{ at time } t \]

\[ = P(q_t = s_i) \]

• Easy to do inductive definition

\[ \forall i \ p_0(i) = \begin{cases} 1 & \text{if } s_i \text{ is the start state} \\ 0 & \text{otherwise} \end{cases} \]

\[ \forall j \ p_{t+1}(j) = P(q_{t+1} = s_j) = \]

\[ \sum_{i=1}^{N} P(q_{t+1} = s_j \land q_t = s_i) = \]
What is $P(q_t = s)$? Clever answer

- For each state $s_i$, define $p_t(i) = \text{Prob. state is } s_i \text{ at time } t = P(q_t = s_i)$
- Easy to do inductive definition
  $$\forall i \quad p_0(i) = \begin{cases} 1 & \text{if } s_i \text{ is the start state} \\ 0 & \text{otherwise} \end{cases}$$
  $$\forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) = \sum_{i=1}^{N} P(q_{t+1} = s_j \land q_t = s_i) = \sum_{i=1}^{N} a_j p_t(i)$$

Remember, $a_j = P(q_{t+1} = s_j \mid q_t = s_i)$

Cost of computing $p_t(i)$ for all states $S_t$ is now $O(t N^2)$
- The stupid way was $O(N^t)$
- This was a simple example
- It was meant to warm you up to this trick, called Dynamic Programming, because HMMs do many tricks like this.

Hidden State

"It’s currently time $t$, and human remains uncrushed. What’s the probability of crushing occurring at time $t + 1$?"

If robot is blind:

We can compute this in advance.

If robot is omnipotent:

(I.E. If robot knows state at time $t$), can compute directly.

Too Easy. We won’t do this.

If robot has some sensors, but incomplete state information …

Hidden Markov Models are applicable!
Hidden State

- The previous example tried to estimate $P(q_t = s_j)$ unconditionally (using no observed evidence).
- Suppose we can observe something that’s affected by the true state.
- Example: Proximity sensors, (tell us the contents of the 8 adjacent squares)

![Diagram](image1.png)

Noisy Hidden State

- Example: Noisy Proximity sensors. (unreliably tell us the contents of the 8 adjacent squares)

![Diagram](image2.png)

Question: what’d be the best Bayes Net structure to represent the Joint Distribution of $(q_0, q_1, q_2, q_3, q_4, O_0, O_1, O_2, O_3, O_4)$?
Hidden Markov Models

Our robot with noisy sensors is a good example of an HMM

- **Question 1: State Estimation**
  - What is \( P(q_T=S_t | O_1, O_2, \ldots, O_T) \)
  - It will turn out that a new cute D.P. trick will get this for us.

- **Question 2: Most Probable Path**
  - Given \( O_1, O_2, \ldots, O_T \), 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 \( O_1, O_2, \ldots, O_T \), 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 (e.g. Reid Simmons / Sebastian Thrun / Sven Koenig)
  - 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.
Some Famous HMM Tasks

Question 1: State Estimation
What is \( P(q_T=S_i \mid O_1O_2...O_t) \)?

Question 2: Most Probable Path
Given \( O_1O_2...O_T \), what is the most probable path that I took?

Question 3: Learning HMMs:
Given \( O_1O_2...O_T \), what is the maximum likelihood HMM that could have produced this string of observations?

Basic Operations in HMMs

For an observation sequence \( O = O_1...O_T \), the three basic HMM operations are:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation: Calculating ( P(q_T=S_i \mid O_1O_2...O_t) )</td>
<td>Forward-Backward</td>
<td>( O(TN^2) )</td>
</tr>
<tr>
<td>Inference: Computing ( Q^* = \text{argmax}_Q P(Q \mid O) )</td>
<td>Viterbi Decoding</td>
<td>( O(TN^2) )</td>
</tr>
<tr>
<td>Learning: Computing ( \lambda^* = \text{argmax}_\lambda P(O \mid \lambda) )</td>
<td>Baum-Welch (EM)</td>
<td>( O(TN^2) )</td>
</tr>
</tbody>
</table>

\( T = \# \) timesteps, \( N = \# \) states
**HMM Notation (from Rabiner’s Survey)**

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

For a particular trial….

Let $T$ be the number of observations

$T$ is also the number of states passed through.

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

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

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

**HMM Formal Definition**

An HMM, $\lambda$, is a 5-tuple consisting of

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

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

- $b_{ij}$ The observation probabilities
  
  \[ P(O_t = k \mid q_t = S_i) = b_{ij}(k) \]

**Here’s an HMM**

Start randomly in state 1 or 2

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

$N = 3$

$M = 3$

$\pi_1 = \frac{1}{2}$

$\pi_2 = \frac{1}{2}$

$\pi_3 = 0$

$a_{11} = 0$

$a_{12} = \frac{1}{3}$

$a_{13} = \frac{2}{3}$

$a_{21} = \frac{2}{3}$

$a_{22} = 0$

$a_{23} = \frac{1}{3}$

$a_{31} = \frac{1}{3}$

$a_{32} = \frac{2}{3}$

$a_{33} = \frac{1}{3}$

$b_1(X) = \frac{1}{2}$

$b_1(Y) = \frac{1}{2}$

$b_1(Z) = 0$

$b_2(X) = 0$

$b_2(Y) = \frac{1}{2}$

$b_2(Z) = \frac{1}{2}$

$b_3(X) = \frac{1}{2}$

$b_3(Y) = 0$

$b_3(Z) = \frac{1}{2}$

**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:

$N = 3$

$M = 3$

$\pi_1 = \frac{1}{2}$

$\pi_2 = \frac{1}{4}$

$\pi_3 = 0$

$a_{11} = 0$

$a_{12} = \frac{1}{3}$

$a_{13} = \frac{2}{3}$

$a_{21} = \frac{2}{3}$

$a_{22} = 0$

$a_{23} = \frac{1}{3}$

$a_{31} = \frac{1}{3}$

$a_{32} = \frac{2}{3}$

$a_{33} = \frac{1}{3}$

$b_1(X) = \frac{1}{2}$

$b_1(Y) = \frac{1}{2}$

$b_1(Z) = 0$

$b_2(X) = 0$

$b_2(Y) = \frac{1}{2}$

$b_2(Z) = \frac{1}{2}$

$b_3(X) = \frac{1}{2}$

$b_3(Y) = 0$

$b_3(Z) = \frac{1}{2}$
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 X and Y

N = 3
M = 3
π₁ = ½
π₂ = ½
π₃ = 0

a₁₁ = ½
a₁₂ = ½
a₁₃ = ½

a₂₁ = ½
a₂₂ = 0
a₂₃ = ½

a₃₁ = ½
a₃₂ = ½
a₃₃ = ½

b₁ (X) = ½
b₁ (Y) = ½
b₁ (Z) = 0

b₂ (X) = 0
b₂ (Y) = ½
b₂ (Z) = ½

b₃ (X) = ½
b₃ (Y) = 0
b₃ (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
π₁ = ½
π₂ = ½
π₃ = 0

a₁₁ = 0
a₁₂ = ½
a₁₃ = ½

a₂₁ = ½
a₂₂ = 0
a₂₃ = ½

a₃₁ = ½
a₃₂ = ½
a₃₃ = ½

b₁ (X) = ½
b₁ (Y) = ½
b₁ (Z) = 0

b₂ (X) = 0
b₂ (Y) = ½
b₂ (Z) = ½

b₃ (X) = ½
b₃ (Y) = 0
b₃ (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:

- \( N = 3 \)
- \( M = 3 \)
- \( \pi_1 = \frac{1}{2} \)
- \( \pi_2 = \frac{1}{2} \)
- \( \pi_3 = 0 \)
- \( a_{11} = 0 \)
- \( a_{12} = \frac{1}{3} \)
- \( a_{13} = \frac{1}{3} \)
- \( a_{21} = \frac{1}{3} \)
- \( a_{22} = 0 \)
- \( a_{23} = \frac{1}{3} \)
- \( a_{31} = \frac{1}{3} \)
- \( a_{32} = \frac{1}{3} \)
- \( a_{33} = \frac{1}{3} \)

- \( b_1 (X) = \frac{1}{2} \)
- \( b_1 (Y) = \frac{1}{2} \)
- \( b_1 (Z) = 0 \)
- \( b_2 (X) = 0 \)
- \( b_2 (Y) = \frac{1}{2} \)
- \( b_2 (Z) = \frac{1}{2} \)
- \( b_3 (X) = \frac{1}{2} \)
- \( b_3 (Y) = 0 \)
- \( b_3 (Z) = \frac{1}{2} \)

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:

- \( N = 3 \)
- \( M = 3 \)
- \( \pi_1 = \frac{1}{2} \)
- \( \pi_2 = \frac{1}{2} \)
- \( \pi_3 = 0 \)
- \( a_{11} = 0 \)
- \( a_{12} = \frac{1}{3} \)
- \( a_{13} = \frac{1}{3} \)
- \( a_{21} = \frac{1}{3} \)
- \( a_{22} = 0 \)
- \( a_{23} = \frac{1}{3} \)
- \( a_{31} = \frac{1}{3} \)
- \( a_{32} = \frac{1}{3} \)
- \( a_{33} = \frac{1}{3} \)

- \( b_1 (X) = \frac{1}{2} \)
- \( b_1 (Y) = \frac{1}{2} \)
- \( b_1 (Z) = 0 \)
- \( b_2 (X) = 0 \)
- \( b_2 (Y) = \frac{1}{2} \)
- \( b_2 (Z) = \frac{1}{2} \)
- \( b_3 (X) = \frac{1}{2} \)
- \( b_3 (Y) = 0 \)
- \( b_3 (Z) = \frac{1}{2} \)

Prob. of a series of observations

What is \( P(O) = P(O_1 O_2 O_3) = P(O_1 = X \land O_2 = X \land O_3 = Z) \)?

Slow, stupid way:

\[
P(O) = \sum_{Q_{Path of length 3}} P(O \land Q) = \sum_{Q_{Path of length 3}} P(O \mid Q) P(Q)
\]

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

How do we compute \( P(O \mid Q) \) for an arbitrary path \( Q \)?
Prob. of a series of observations

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

Slow, stupid way:

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

\[ = \sum_{Q \in \text{Paths of length 3}} P(O|Q)P(Q) \]

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

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

\[ P(O) = P(q_1, q_2, q_3) \] (chain rule)

\[ = P(q_1) P(q_2|q_1) P(q_3|q_2) \] (chain)

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

\[ \frac{1}{2} \times \frac{2}{3} \times \frac{1}{3} = \frac{1}{9} \]

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

Given observations $O_1, O_2, \ldots, O_T$

Define

\[ \alpha_i(t) = P(O_1, O_2, \ldots, O_t | q_t = S_i \land \lambda) \quad \text{where } 1 \leq t \leq T \]

\[ \alpha_i(i) = \text{Probability that, in a random trial,} \]

- 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

\[
\alpha_t(i) = P(O_1 O_2 \ldots O_t \land q_t = S_t \mid \lambda) \quad (\alpha_t(i) \text{ can be defined stupidly by considering all paths length } t).
\]

\[
\alpha_t(i) = P(O_i \land q_i = S_i) = P(q_i = S_i \mid O_i) = P(q_i = S_j) \\
\alpha_{t+1}(j) = P(O_{t+1} \ldots O_T \land q_{t+1} = S_j)
\]

**How?**

\[
\alpha_{t+1}(j) = \sum_{i \in \mathcal{I}} P(O_{t+1} \ldots O_T \land q_{t+1} = S_j \mid O_{t+1} \land q_{t+1} = S_j) \alpha_t(i)
\]

**Easy Question**

We can cheaply compute

\[
\alpha_t(i) = P(O_1 O_2 \ldots O_T \land q_t = S_t) \\
(\text{How}) \text{ can we cheaply compute } P(O_1 O_2 \ldots O_T) \ ?
\]

(How) can we cheaply compute

\[
P(q_t = S_j \mid O_1 O_2 \ldots O_t)
\]
Easy Question
We can cheaply compute
\[ \alpha_t(i) = P(O_1O_2...O_t \land q_t = S_i) \]
(How) can we cheaply compute
\[ P(O_1O_2...O_t) = \sum_{i=1}^{N} \alpha_t(i) \]
(How) can we cheaply compute
\[ P(q_t = S_i | O_1O_2...O_t) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)} \]

Most probable path given observations
What's most probable path given \( O_1O_2...O_T \), i.e.
What is \( \arg\max_Q P(O_1O_2...O_T) \)?
Slow, stupid answer :
\[ \arg\max_Q P(O_1O_2...O_T) = \arg\max_Q \frac{P(O_1O_2...O_T | Q)P(Q)}{P(O_1O_2...O_T)} = \arg\max_Q P(O_1O_2...O_T | Q)P(Q) \]

Efficient MPP computation
We're going to compute the following variables:
\[ \delta_t(i) = \max_{q_1q_2..q_{t-1}} P(q_1q_2..q_{t-1} \land q_t = S_i \land O_1..O_t) \]
= 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...O_t

DEFINE: \( mpp_t(i) = \) that path
So:
\[ \delta_t(i) = \text{Prob}(mpp_t(i)) \]

The Viterbi Algorithm
\[ \delta_t(i) = \max_{q_1q_2..q_{t-1}} P(q_1q_2..q_{t-1} \land q_t = S_i \land O_1O_2..O_t) \]
\[ mpp_t(i) = \max_{q_1q_2..q_{t-1}} P(q_1q_2..q_{t-1} \land q_t = S_i \land O_1O_2..O_t) \]
\[ \delta_t(i) = \text{one choice} \ P(q_t = S_i \land O_t) = P(q_t = S_i) \text{Prob}(O_t | q_t = S_i) = \pi_{b_t}(O_t) \]

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 \( mpp_t+1(j) \)?
\[ mpp_t(1) \rightarrow \delta_t(1) \rightarrow S_1 \]
\[ mpp_t(2) \rightarrow \delta_t(2) \rightarrow S_2 \]
\[ mpp_t(n) \rightarrow \delta_t(n) \rightarrow S_n \]
\[ q_t \rightarrow \delta_t+1(q_{t+1}) \rightarrow S_i \]
The Viterbi Algorithm

The most probable path with last two states $S_i$, $S_j$ is the most probable path to $S_i$, followed by transition $S_i \rightarrow S_j$

What is the prob of that path?

$$\delta_t(i) \times P(S_i \rightarrow S_j \land O_{t+1} | \lambda) = \delta_t(i) a_{ij} b_j(O_{t+1})$$

SO: The most probable path to $S_j$ has $S_i$ as its penultimate state, where $i^* = \text{argmax}_i \delta_t(i) a_{ij} b_j(O_{t+1})$

Summary:

$$\delta_{t+1}(j) = \delta_t(i^*) a_{ij} b_j(O_{t+1})$$

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. \( O_1 O_2 \ldots O_T \) with a big “T”.

Observations previously in lecture \( O_1 O_2 \ldots O_T \)

Observations in the next bit \( O_1 O_2 \ldots O_T \)

Inferring an HMM

Remember, we’ve been doing things like

\[
P(O_1 O_2 \ldots O_T | \lambda)
\]

That “\( \lambda \)” is the notation for our HMM parameters.

Now we have some observations and we want to estimate \( \lambda \) from them.

AS USUAL: We could use

(i) MAX LIKELIHOOD

\[
\lambda = \text{argmax } P(O_1 \ldots O_T | \lambda)
\]

(ii) BAYES

Work out \( P(\lambda | O_1 \ldots O_T) \)

and then take \( E[\lambda] \) or max \( P(\lambda | O_1 \ldots O_T) \)

Max likelihood HMM estimation

Define

\[
\gamma_t(i) = P(q_t = S_i | O_1O_2\ldots O_T, \lambda)
\]

\[
\epsilon_t(i,j) = P(q_t = S_i \land q_{t+1} = S_j | O_1O_2\ldots O_T, \lambda)
\]

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

(Details in Rabiner paper)

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

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

HMM estimation

Notice

\[
\sum_{t=1}^{T-1} \gamma_t(i) = \left( \frac{\text{expected frequency}}{i \rightarrow j} \right)
\]

\[
\sum_{t=1}^{T-1} \epsilon_t(i,j) = \left( \frac{\text{expected frequency}}{i} \right)
\]

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

We can re-estimate

\[
a_j \leftarrow \frac{\sum \epsilon_t(i,j)}{\sum \gamma_t(i)}
\]

We can also re-estimate

\[
b_j(O_k) \leftarrow \ldots
\]

(See Rabiner)
We want $a_{ij}^{\text{new}} = \text{new estimate of } P(q_{t+1} = s_j \mid q_t = s_i)$

\[
= \frac{\text{Expected \# transitions } i \to j \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T}{\sum_{k=1}^{N} \text{Expected \# transitions } i \to k \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T} \\
= \frac{\sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}{\sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_k, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}
\]

We want $a_{ij}^{\text{new}} = \text{new estimate of } P(q_{t+1} = s_j \mid q_t = s_i)$

\[
= \frac{\text{Expected \# transitions } i \to j \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T}{\sum_{k=1}^{N} \text{Expected \# transitions } i \to k \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T} \\
= \frac{\sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}{\sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_k, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}
\]

\[
= \frac{\sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}{\sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_k, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}
\]

= What?

where $S_{ij} = \sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i, O_1, \cdots O_T \mid \lambda^{\text{old}})$
We want \( a_{ij}^{\text{new}} = \text{new estimate of } P(q_{t+1} = s_j | q_t = s_i) \)

\[
\sum_{k=1}^{N} \frac{\text{Expected } i \rightarrow j | \lambda^{\text{old}}, O_1, O_2, \cdots O_T}{\sum_{k=1}^{N} \text{Expected } i \rightarrow k | \lambda^{\text{old}}, O_1, O_2, \cdots O_T}
\]

\[
= \frac{\sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i | \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}{\sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_k, q_t = s_i | \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}
\]

\[
= \frac{S_{ij}}{\sum_{k=1}^{N} S_{ik}} \text{ where } S_{ij} = \sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i, O_1, \cdots O_T | \lambda^{\text{old}})
\]

\[
= a_{ij} \sum_{t=1}^{T} \alpha_t(i) \beta_{t+1}(j) b_j(O_{t+1})
\]

---

**EM for HMMs**

If we knew \( \lambda \) we could estimate EXPECTATIONS of quantities such as:
- Expected number of times in state \( i \)
- Expected number of transitions \( i \rightarrow j \)

If we knew the quantities such as:
- Expected number of times in state \( i \)
- Expected number of transitions \( i \rightarrow j \)
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 $\lambda$ estimate $\lambda(0)$, $k=0$
3. $k = k+1$
4. Given $O_1 \ldots O_T$, $\lambda(k)$ compute $\gamma_t(i), \epsilon_t(i,j)$ $\forall 1 \leq t \leq T, \forall 1 \leq i \leq N, \forall 1 \leq j \leq N$
5. Compute expected freq. of state $i$, and expected freq. $i \rightarrow j$
6. Compute new estimates of $a_{ij}, b_j(k), \pi_i$ accordingly. Call them $\lambda(k+1)$
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 $\lambda(0)$
• 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”].