# Markov Chain and Hidden Markov Models for Speech Recognition Systems

Siddhartha Saxena, Siddharth Mittal, Ankit Bharadwaj

Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

October 22, 2016

# Introduction to Markov Chains

**Definition**

A probabilistic model describing a sequence of possible events in which the probability of each event depends on the states attained in the previous event.

# Introduction to Markov chains

- A Markov chain is a sequence $X_0, X_1, X_2, \ldots$ of random variables that indicate states $1, 2, 3, \ldots, m$ such that there is a probability $p_{ij}$ that $X_{n+1} = i$ given that $X_n = j$.

# Introduction to Markov chains

- A Markov chain is a sequence $X_0, X_1, X_2, \ldots$ of random variables that indicate states $1, 2, 3, \ldots, m$ such that there is a probability $p_{ij}$ that $X_{n+1} = i$ given that $X_n = j$.
- $p_{ij}$ are called *transition probabilities* whose sum for each fixed i is 1. We can write the transition probabilities in matrix form as follows:

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1m} \\ p_{21} & \cdots & p_{2m} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{bmatrix}$$

$P$ is called the Transition matrix

# Introduction to Markov chains

- A Markov chain is a sequence $X_0, X_1, X_2, \ldots$ of random variables that indicate states $1, 2, 3, \ldots, m$ such that there is a probability $p_{ij}$ that $X_{n+1} = i$ given that $X_n = j$.
- $p_{ij}$ are called *transition probabilities* whose sum for each fixed i is 1. We can write the transition probabilities in matrix form as follows:

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1m} \\ p_{21} & \cdots & p_{2m} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{bmatrix}$$

  $P$ is called the Transition matrix
- A state vector X is changed to a new state vector X' using the transition matrix as follows:

$$X' = PX$$

# Let's look at an example..

- A boy, a girl, and a dog are playing with a ball.
  The boy throws the ball to the girl 2/3 of the time and to the dog 1/3 of the time.
  The girl throws the ball to the boy 1/2 of the time and to the dog 1/2 of the time.
  The dog brings the ball to the girl all of the time.

# Let's look at an example..

- A boy, a girl, and a dog are playing with a ball.
  The boy throws the ball to the girl 2/3 of the time and to the dog 1/3 of the time.
  The girl throws the ball to the boy 1/2 of the time and to the dog 1/2 of the time.
  The dog brings the ball to the girl all of the time.
- We can turn the situation into a matrix equation.
  Let the probabilities that the boy, the girl, and the dog have the ball at time n be $b_n$, $g_n$, and $d_n$, respectively.
  The initial probabilities $b_0$, $g_0$, and $d_0$ are three non-negative real numbers that sum to 1.

- The probabilities satisfy the following matrix equation:

$$\begin{bmatrix} b_{n+1} \\ g_{n+1} \\ d_{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 1/2 & 0 \\ 2/3 & 0 & 1 \\ 1/3 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} b_n \\ g_n \\ d_n \end{bmatrix}$$

- $$\begin{bmatrix} 0 & 1/2 & 0 \\ 2/3 & 0 & 1 \\ 1/3 & 1/2 & 0 \end{bmatrix}$$

  is the transition matrix for given example.

# Another example..

- A famous Markov chain is the so-called "drunkard's walk", a random walk on the number line where, at each step, the position may change by $+1$ or $-1$ with equal probability. From any position there are two possible transitions, to the next or previous integer.

# Another example..

- A famous Markov chain is the so-called "drunkard's walk", a random walk on the number line where, at each step, the position may change by $+1$ or $-1$ with equal probability. From any position there are two possible transitions, to the next or previous integer.
- The transition probabilities depend only on the current position, not on the manner in which the position was reached. For example, the transition probabilities from 5 to 4 and 5 to 6 are both 0.5, and all other transition probabilities from 5 are 0. These probabilities are independent of whether the system was previously in 4 or 6.

# Hidden Markov Model

## Definition

- A hidden markov model is a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols.

# Explanation through example

- Consider the example of weather on any given day. Assume that each 'day' corresponds to a state. For simplicity, we assume that only three types of weather are possible namely, sunny, rainy and foggy. Take 1st order Markov approximation that the weather on day n is decided entirely by weather on day n-1.

# Explanation through example

- Consider the example of weather on any given day. Assume that each ’day’ corresponds to a state. For simplicity, we assume that only three types of weather are possible namely, sunny, rainy and foggy. Take 1st order Markov approximation that the weather on day n is decided entirely by weather on day n-1.
- Let the transition probabilities be given by the corresponding table:

| today’s weather | tomorrow’s weather | | |
|---|---|---|---|
| | sunny | rainy | foggy |
| sunny | 0.8 | 0.05 | 0.15 |
| rainy | 0.2 | 0.6 | 0.2 |
| foggy | 0.2 | 0.3 | 0.5 |

- Now consider that you are locked in a room for several days and the only clue you get for the weather outside is whether the person carrying your daily meal carries an umberella or not. The corresponding probability table is given by:

| weather | probability of umberella |
|---|---|
| sunny | 0.1 |
| rainy | 0.8 |
| foggy | 0.3 |

- Now consider that you are locked in a room for several days and the only clue you get for the weather outside is whether the person carrying your daily meal carries an umberella or not. The corresponding probability table is given by:

| weather | probability of umberella |
|---|---|
| sunny | 0.1 |
| rainy | 0.8 |
| foggy | 0.3 |

- In this example, the points to be noted are 1)The actual weather is hidden from you. 2)The process is doubly stochastic as any outcome is achieved with two probability distributions 3)The observed symbols are presence or absence of umbrella on any day.

# Elements of HMM

- There is a finite number, say N, of states in our HMM. In the considered example, there were 3 states, i.e the different types of weather.

# Elements of HMM

- There is a finite number, say N, of states in our HMM. In the considered example, there were 3 states, i.e the different types of weather.
- At each clock time t, a new state is entered based on the transition probability distribution which depends on the previous state(by the Markovian property). Inn our example, each new day was the new clock time and the table 1 was transition probabilities between states.

# Elements of HMM

- There is a finite number, say N, of states in our HMM. In the considered example, there were 3 states, i.e the different types of weather.
- At each clock time t, a new state is entered based on the transition probability distribution which depends on the previous state(by the Markovian property). Inn our example, each new day was the new clock time and the table 1 was transition probabilities between states.
- After each transition is made, an observation output symbol is produced according to a probability distribution(Table 2) which depends on the current state and remains fixed for that state regardless of when or how the state is achieved.

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T = length of observation sequence (number of clock times)

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T = length of observation sequence (number of clock times)
- N = number of states in the model

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T = length of observation sequence (number of clock times)
- N = number of states in the model
- M = number of possible observation symbols

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T = length of observation sequence (number of clock times)
- N = number of states in the model
- M = number of possible observation symbols
- Q = $\{q_1, q_2, ...., q_N\}$, states

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T = length of observation sequence (number of clock times)
- N = number of states in the model
- M = number of possible observation symbols
- Q $= \{q_1, q_2, ...., q_N\}$, states
- V $= \{v_1, v_2, ..., v_M\}$, discrete set of possible observations

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T $=$ length of observation sequence (number of clock times)
- N $=$ number of states in the model
- M $=$ number of possible observation symbols
- Q $= \{q_1, q_2, ...., q_N\}$, states
- V $= \{v_1, v_2, ..., v_M\}$, discrete set of possible observations
- A $= \{a_{ij}\}$, $a_{ij} = Pr(q_j \text{ at } t+1 | q_i \text{ at } t)$, state transition probability distribution

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T $=$ length of observation sequence (number of clock times)
- N $=$ number of states in the model
- M $=$ number of possible observation symbols
- Q $= \{q_1, q_2, ...., q_N\}$, states
- V $= \{v_1, v_2, ..., v_M\}$, discrete set of possible observations
- A $= \{a_{ij}\}$, $a_{ij} = Pr(q_j \text{ at } t+1|q_i \text{ at } t)$, state transition probability distribution
- B $= \{b_j(k)\}$,$b_j(k) = Pr(v_k \text{ at } t|q_j \text{ at } t)$, observation symbol probability distribution in state j

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T $=$ length of observation sequence (number of clock times)
- N $=$ number of states in the model
- M $=$ number of possible observation symbols
- Q $= \{q_1, q_2, ...., q_N\}$, states
- V $= \{v_1, v_2, ..., v_M\}$, discrete set of possible observations
- A $= \{a_{ij}\}$, $a_{ij} = Pr(q_j \text{ at } t+1|q_i \text{ at } t)$, state transition probability distribution
- B $= \{b_j(k)\}$,$b_j(k) = Pr(v_k \text{ at } t|q_j \text{ at } t)$, observation symbol probability distribution in state j
- $\pi = \{\pi_i\}, \pi_i = Pr(q_i \text{ at } t = 1)$, initial state distribution

# Formal Model

We now formally define the following model notation for discrete observation HMM:

- T $=$ length of observation sequence (number of clock times)
- N $=$ number of states in the model
- M $=$ number of possible observation symbols
- Q $= \{q_1, q_2, ...., q_N\}$, states
- V $= \{v_1, v_2, ..., v_M\}$, discrete set of possible observations
- A $= \{a_{ij}\}$, $a_{ij} = Pr(q_j \text{ at } t+1 | q_i \text{ at } t)$, state transition probability distribution
- B $= \{b_j(k)\}$,$b_j(k) = Pr(v_k \text{ at } t | q_j \text{ at } t)$, observation symbol probability distribution in state j
- $\pi = \{\pi_i\}, \pi_i = Pr(q_i \text{ at } t = 1)$, initial state distribution
- O $= \{O_1, O_2, ..., 0_T\}$, observation sequence

# Use of HMMs in speech recognition

**Isolated Word Model**

# Use of HMMs in speech recognition

## Isolated Word Model

1. Spectral Representation of words and Vector Quantization
   - The speech signal received needs to be represented in such a way that it is linguistically meaningful as well as computationally favourable. For this purpose the received speech signal is represented by its spectrum, i.e. frequency distribution of intensity. Various models such as bank of filter and linear prediction are used for the spectral analysis of speech signal received. This gives us a continuous representation of a word

# Use of HMMs in speech recognition

## Isolated Word Model

1. Spectral Representation of words and Vector Quantization
   - The speech signal received needs to be represented in such a way that it is linguistically meaningful as well as computationally favourable. For this purpose the received speech signal is represented by its spectrum, i.e. frequency distribution of intensity. Various models such as bank of filter and linear prediction are used for the spectral analysis of speech signal received. This gives us a continuous representation of a word
   - Then we can convert it into discrete objects like phonemes using a nearest neighbor model whee we compare the given phonemes with the output of the spectral analysis and give the most similar or "nearest neighbour"as the output.

# Isolated word model contd.

2. Building an HMM for each word
    - We assume that we are having a vocabulary of V words and for each word we build a markov model. Now here each word is made up of occurrences $O_1, O_2, ...O_N$ each being an element of the codebook

# Isolated word model contd.

2. Building an HMM for each word
    - We assume that we are having a vocabulary of V words and for each word we build a markov model. Now here each word is made up of occurrences $O_1, O_2, ...O_N$ each being an element of the codebook
    - Now we can define different number of states in our model. Here we consider the number of states as the number of phonemes that occur in the word and each state has different probabilities for different in-dices of the codebook. For example in a codebook of phonemes, there are 44 different probabilities associate with a state.

# Isolated word model contd.

2. Building an HMM for each word
   - We assume that we are having a vocabulary of V words and for each word we build a markov model. Now here each word is made up of occurrences $O_1, O_2, ...O_N$ each being an element of the codebook
   - Now we can define different number of states in our model. Here we consider the number of states as the number of phonemes that occur in the word and each state has different probabilities for different in-dices of the codebook. For example in a codebook of phonemes, there are 44 different probabilities associate with a state.
   - In order to train our model we use the training data for a single word spoken by several speakers. Then we have to train our model parameters for transformation from one state to another using the problem number three.

# Isolated Word Model contd.

- To give an example consider the word "sit", the phonemes can be /s/,/i/,/t/ or /z/,/e/,/t/ or anything else with varying probabilities depending on the speaker and there will be a different probability of occurrence of /e/ in two different states which we need to find out while training which is done by the EM Algorithm.

# Isolated Word Model contd.

- To give an example consider the word "sit", the phonemes can be /s/,/i/,/t/ or /z/,/e/,/t/ or anything else with varying probabilities depending on the speaker and there will be a different probability of occurrence of /e/ in two different states which we need to find out while training which is done by the EM Algorithm.
- Now during test time what we do is that we convert the spoken word or words into features using Vector Quantization and then concentrate on individual words first. We then evaluate the probabilities of it being any of the word in our vocabulary using the HMMs we trained but computationally, this process is very expensive hence we shall introduce the Forward-Backward algorithm to overcome it.

# Isolated Word Model contd.

- To give an example consider the word "sit", the phonemes can be /s/,/i/,/t/ or /z/,/e/,/t/ or anything else with varying probabilities depending on the speaker and there will be a different probability of occurrence of /e/ in two different states which we need to find out while training which is done by the EM Algorithm.
- Now during test time what we do is that we convert the spoken word or words into features using Vector Quantization and then concentrate on individual words first. We then evaluate the probabilities of it being any of the word in our vocabulary using the HMMs we trained but computationally, this process is very expensive hence we shall introduce the Forward-Backward algorithm to overcome it.
- Then we can also use the constrains like it being a meaningful word as well as the syntactic constraints of grammar and semantic constraints.

# The three problems in HMMs

1. The Evaluation Problem
2. The Sequence Choosing Problem
3. Finding the Model Parameters

# The Evaluation Problem

- The evaluation problem is that given a model and the sequence of observations, how do we compute the probability that the observed sequence was produced by the model.

# The Evaluation Problem

- The evaluation problem is that given a model and the sequence of observations, how do we compute the probability that the observed sequence was produced by the model.
- Given the observations $O_1, O_2, ...., O_T$ and states $q_1, q_2, ..., q_T$ we have we need to find the probability of occurrence of this observation sequence i.e. $P(O|\lambda)$ where $\lambda$ signifies the model parameters.

# The Evaluation Problem

- The evaluation problem is that given a model and the sequence of observations, how do we compute the probability that the observed sequence was produced by the model.
- Given the observations $O_1, O_2, ...., O_T$ and states $q_1, q_2, ..., q_T$ we have we need to find the probability of occurrence of this observation sequence i.e. $P(O|\lambda)$ where $\lambda$ signifies the model parameters.
- The straightforward way of doing this is through enumerating every possible state sequence of length T (The number of observations).

- We consider one such state sequence $Q = q_1 q_2 ... q_T$
  The probability of observation sequence $O$ for the above state sequence is:
  $P(O|Q,\lambda) = \prod_{t=1}^{t=T} P(O_t|q_t,\lambda)$
  We assume independence of observations

- We consider one such state sequence $Q = q_1 q_2 ... q_T$
  The probability of observation sequence $O$ for the above state sequence is:
  $P(O|Q, \lambda) = \prod_{t=1}^{t=T} P(O_t|q_t, \lambda)$
  We assume independence of observations
- The joint probability that $O$ and $Q$ occur simultaneously is simply:
  $P(O, Q|\lambda) = P(O|Q, \lambda)P(Q|\lambda)$

- We consider one such state sequence $Q = q_1 q_2 ... q_T$
  The probability of observation sequence $O$ for the above state sequence is:
  $P(O|Q,\lambda) = \prod_{t=1}^{t=T} P(O_t|q_t,\lambda)$
  We assume independence of observations
- The joint probability that $O$ and $Q$ occur simultaneously is simply:
  $P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda)$
- Then $P(O|\lambda)$ is obtained by summing this joint probability over all possible state sequences, giving:
  $P(O|\lambda) = \sum_{allQ} P(O|Q,\lambda)P(Q|\lambda)$
  $= \sum_{q_1,..,q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} ... a_{q_{T-1} q_T} b_{q_T}(O_T)$

- We consider one such state sequence $Q = q_1 q_2 ... q_T$
  The probability of observation sequence $O$ for the above state sequence is:
  $P(O|Q,\lambda) = \prod_{t=1}^{t=T} P(O_t|q_t,\lambda)$
  We assume independence of observations
- The joint probability that $O$ and $Q$ occur simultaneously is simply:
  $P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda)$
- Then $P(O|\lambda)$ is obtained by summing this joint probability over all possible state sequences, giving:
  $P(O|\lambda) = \sum_{allQ} P(O|Q,\lambda)P(Q|\lambda)$
  $= \sum_{q_1,..,q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} ... a_{q_{T-1} q_T} b_{q_T}(O_T)$
- This process involves roughly $2T * N^T$ calculations, which is computationally unfeasible.
  (There are $N^T$ possible states and each state involves roughly $2T$ calculations)

# Way around this

In order to reduce the complexity, the forward-backward procedure is used. It has three steps which give a complexity of the order of $N^2T$, which is much better.

# Way around this

In order to reduce the complexity, the forward-backward procedure is used. It has three steps which give a complexity of the order of $N^2 T$, which is much better.

- The Forward Probability (The probability of the partial observation sequence $O_1, O_2, .., O_t$ and state $S_i$ at time t):

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, q_t = S_i|\lambda)$$

for $1 \leq i \leq N$

# Way around this

In order to reduce the complexity, the forward-backward procedure is used. It has three steps which give a complexity of the order of $N^2T$, which is much better.

- The Forward Probability (The probability of the partial observation sequence $O_1, O_2, .., O_t$ and state $S_i$ at time t):

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, q_t = S_i|\lambda)$$

  for $1 \leq i \leq N$

- Now we can calculate $\alpha_t's$ inductively as

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

# Way around this

In order to reduce the complexity, the forward-backward procedure is used. It has three steps which give a complexity of the order of $N^2T$, which is much better.

- The Forward Probability (The probability of the partial observation sequence $O_1, O_2, .., O_t$ and state $S_i$ at time t):

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, q_t = S_i|\lambda)$$

  for $1 \leq i \leq N$
- Now we can calculate $\alpha_t's$ inductively as

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

- Finally $P(O|\lambda)$ is the sum of all $\alpha_T(i)'s$:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

- In a manner similar to forward probability variable, we define a backward probability variable as
  $\beta_t(i) = P(O_{t+1}O_{t+2}...O_T|q_t = S_i, \lambda)$
  i.e., the probability of the partial observation sequence from t+1 to the end, given state $S_i$ at time $t$ and the model $\lambda$.

- In a manner similar to forward probability variable, we define a backward probability variable as
  $\beta_t(i) = P(O_{t+1}O_{t+2}...O_T|q_t = S_i, \lambda)$
  i.e., the probability of the partial observation sequence from t+1 to the end, given state $S_i$ at time $t$ and the model $\lambda$.
- The following inductive step is used for finding $\beta$'s.

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

# Problem 2

- The problem is to find the optimal state sequence associated with the given observation sequence.
- The difficulty lies with the definition of optimal state sequence; i.e.,there are several possible optimality criteria.
- As an example it may be optimal to chose states $q_t$ which are individually most likely.

- Lets solve the problem which uses the above optimality criteria.
- We define a variable $\gamma_t(i) = P(q_t = S_i|O, \lambda)$ which is the probability of being in state $S_i$ at time $t$ given the observation sequence $O$ and the model $\lambda$

- Lets solve the problem which uses the above optimality criteria.
- We define a variable $\gamma_t(i) = P(q_t = S_i|O, \lambda)$ which is the probability of being in state $S_i$ at time $t$ given the observation sequence $O$ and the model $\lambda$
- $\gamma_t$ can be expressed in terms of forward-backward variables, i.e., $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{i=N} \alpha_t(i)\beta_t(i)}$ since $\alpha_t(i)$ accounts for the partial observation sequence $O_1, O_2, .., O_t$ and state $S_i$ at time t, while $\beta_t(i)$ accounts for the remaining observation sequence given state $S_i$ at time t

- Lets solve the problem which uses the above optimality criteria.
- We define a variable $\gamma_t(i) = P(q_t = S_i|O, \lambda)$ which is the probability of being in state $S_i$ at time $t$ given the observation sequence $O$ and the model $\lambda$
- $\gamma_t$ can be expressed in terms of forward-backward variables, i.e.,
  $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{i=N} \alpha_t(i)\beta_t(i)}$
  since $\alpha_t(i)$ accounts for the partial observation sequence $O_1, O_2, .., O_t$ and state $S_i$ at time t, while $\beta_t(i)$ accounts for the remaining observation sequence given state $S_i$ at time t
- Using $\gamma_t(i)$ we can solve for individually most likely state $q_t$ at time $t$ as
  $q_t = argmax[\gamma_t(i)] \qquad 1 \leq i \leq N$

# Finding the parameters $\lambda$

This is by far the toughest problem in HMMs. We need an algorithm to find the optimal $a_{ij}$ and $b_i(O_t)$ for different i,j and t. In order to do this we use the popular Expectation Maximization algorithm or EM algorithm.

# Finding the parameters $\lambda$

This is by far the toughest problem in HMMs. We need an algorithm to find the optimal $a_{ij}$ and $b_i(O_t)$ for different i,j and t. In order to do this we use the popular Expectation Maximization algorithm or EM algorithm.

EM Algorithm is a two step algorithm which is used in many Probabilistic machine learning models . Below is the general formulation of EM algorithm

# Finding the parameters $\lambda$

This is by far the toughest problem in HMMs. We need an algorithm to find the optimal $a_{ij}$ and $b_i(O_t)$ for different i,j and t. In order to do this we use the popular Expectation Maximization algorithm or EM algorithm.

EM Algorithm is a two step algorithm which is used in many Probabilistic machine learning models . Below is the general formulation of EM algorithm

We assume the model parameters are generated from some probability distribution. Here we can assume that the $a_{ij}$ are generated from several mutinomial distributions for each i as they are discrete whereas we can think of the $b_i(O_t)$ to be coming from a normal distribution for continuous data or from multinomial for the codebooks we shall use.

# Finding the parameters $\lambda$

This is by far the toughest problem in HMMs. We need an algorithm to find the optimal $a_{ij}$ and $b_i(O_t)$ for different i,j and t. In order to do this we use the popular Expectation Maximization algorithm or EM algorithm.

EM Algorithm is a two step algorithm which is used in many Probabilistic machine learning models . Below is the general formulation of EM algorithm

We assume the model parameters are generated from some probability distribution. Here we can assume that the $a_{ij}$ are generated from several mutinomial distributions for each i as they are discrete whereas we can think of the $b_i(O_t)$ to be coming from a normal distribution for continuous data or from multinomial for the codebooks we shall use.

Now these are "latent features"of our model. We represent the state sequence with $Q$ and the parameters which control their distributions which are represented as $\lambda$

# EM Algorithm cont.

**The Expectation Step**: First we randomly initialize the parameters $\lambda$ and then at every iteration we estimate them with the posterity distribution

$$P(Q|O,\lambda^{old}) \propto P(O|Q,\lambda^{old}) * P(Q|\lambda^{old})$$

# EM Algorithm cont.

**The Expectation Step**: First we randomly initialize the parameters $\lambda$ and then at every iteration we estimate them with the posterity distribution

$$P(Q|O,\lambda^{old}) \propto P(O|Q,\lambda^{old}) * P(Q|\lambda^{old})$$

**The Maximization Step**: Now instead of dong MLE on the data, we use the $P(Q|O,\lambda^{old})$ to maximize the likelihood of the $\mathbb{E}(P(O,Q|\lambda))$ which can be proven to be a tight lower bound on $P(O|\lambda)$ hence we can increase it indirectly. Here we maximize the log of this lower bound to simplify the calculations.

$$Q(\lambda_{old},\lambda) = \sum_{z} P(Q|O,\lambda_{old}) log\{p(O,Q|\lambda)\}$$

# For Further Reading I

- M. Erickson *Pearls of Discrete Mathematics*.
- Rabiner-Zheung An introduction to Hidden Markov Model