Hidden Markov Model (HMM)

Saturday, November 24, 2018 22:46

For the glory of God

Introduction

· According to Wikt. Hidden Markov Model (HMM) is defined as a statistical Markov model in which the system being modeled is assumed to be a Markov process with latent states.

· We actually discussed about two words highlighted above. So, what are they?

a) Latent variable

- It is a variable that cannot be observed directly but its values can be inferred by taking other measurement.

e.g. Intelligence, responsibility in EM algorithm, and so forth

b) Markov Model

- The concept of modeling sequences of random events using states and transition between states is known as a Markov chain.
- In the Markov chain, it is satisfied that;

$P(X_k \mid X_{k-1}, X_{k-2}, \cdots, X_1, X_0) = P(X_k \mid X_{k-1})$ → It says the future state depends only on the current state.

- Markov model is a model that follows the characteristic of Markov chain in a changing system.

· Then, What is the HMM? and how does it work?

: Before we dive into the answers, let's a little bit more talk about Motivation.

Motivation

· Perhaps, a simple model is that observations are assumed to be independent and identically distributed (IID).

· However, in most cases in reality, the observations are actually dependent.

- For example, yesterday weather has to be related to today's weather.

· Thankfully, the Markov model demonstrates that; (under the law of large numbers)

The nth observation in a chain of observations is only influenced by the n-1 th observation

· But, What if the nth observation in a chain of observations is influenced by a corresponding latent variable?

e.g. Let's think about the problem where we try to infer the past behavior through a daily record of expenses.

S Latent (Hidden) States : The past behavior such as study or hang out

observations : The daily record of expenses such as credit card history

We talk about discrete latent variable.

(If it's continuous, called as Kalman filter.)

· In this example, the HMM will enable us to infer the hidden states through the observations based on probability calculations.

What is Hidden Markov Model ?

· A HMM is considered as a generalization of a mixture model where the hidden variables are related through a Markov process rather than independent of each other.

· A HMM can be considered as a tool for representing probability distributions over sequences of observations.

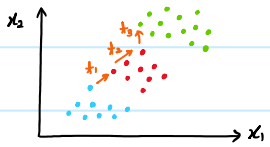
· A HMM gets its name from two defining properties ;

- It assumes that the observation at time t was generated by some process whose state S_t is hidden from the observer.
- It assumes that the state of the hidden process satisfies the Markov chain property.

: That is, given the value of S_{t-1}, the current state S_t is independent of all the states prior to $t-1$.

· In particular, the HMM has been widely used for modeling time series data.

e.g. Let's say that we have the following dataset ;



: It is obvious that we were able to cluser the dataset with three different cluster sets.

↳ Yes, it was perfectly clustered in the space $\mathbb{R}^2$

: What if we want to know the pattern with different time series such as $t_1 \rightarrow t_2 \rightarrow t_3$?

↳ they may have different clusters as time goes.

· The HMM is able to handle this type of problem ;

- For this reason, the HMM is sometimes called as dynamic clustering !

· In summary,

- A model with IID assumption may be the simplest model.
- However, in reality, most are dependent.
- Thanks to Markov, he demonstrated that the future state depends only on the current state.
- In many cases, however, the states we're interested in are hidden. In this sense, we can't observe them directly.
- A HMM allows us to infer the states by investigating the observations.

Difference between Markov Model (MM) and Hidden Markov Model (HMM)

· Before we talk about how the HMM does work, let's first talk about the difference MM vs. HMM.

· Basically,

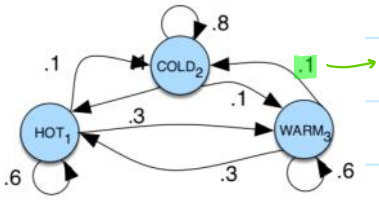
- In MM, the state is directly visible ; therefore, the state transition probabilities are the only parameter.

- In HMM, the state is not directly visible ; but the observations are known.

· Let us take a weather example.

a) Markov Model for weather

- It's as if to predict tomorrow's weather you could examine today's weather but you were not allowed to look at yesterday's weather.



↳ 10% probability to happen from $Warm_3$ to $Cold_2$

; where $\pi = [0.1, 0.7, 0.2]$

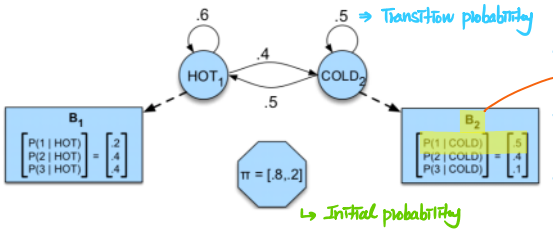
- A start distribution π is required ; for example, setting $\pi = [0.1, 0.7, 0.2]$ for this case.

e.g. Probability 0.1 of starting in state 1 (HOT)

b) Hidden Markov Model

- Let's say that you are such a weather scientist.

- You may have to figure out the weather two years ago. The only information given to you is about Ice cream consumption records. ↳ This will be the hidden variable for this case.



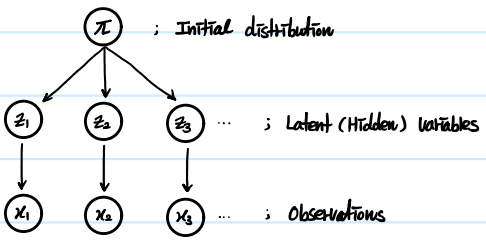
→ Emission probability

e.g. When the weather is cold, the probability having one icecream is 0.5

- Using the observation (records), the HMM enables us to figure out the weather at the day.

· Let us take another example with Gaussian Mixture model ↳ We'll get there soon !

- In terms of the GMM,



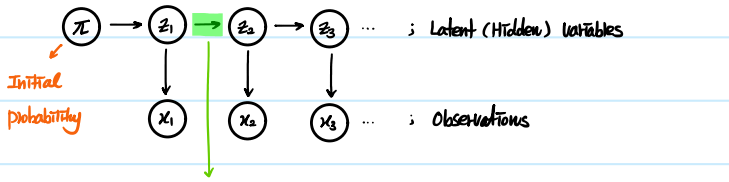
; Initial distribution

; Latent (Hidden) variables

; Observations

; As can be seen, the hidden variables are independent.

- In terms of time series data for GMM,



z_2 latent variable is influenced by z_1 latent variable as time goes by.

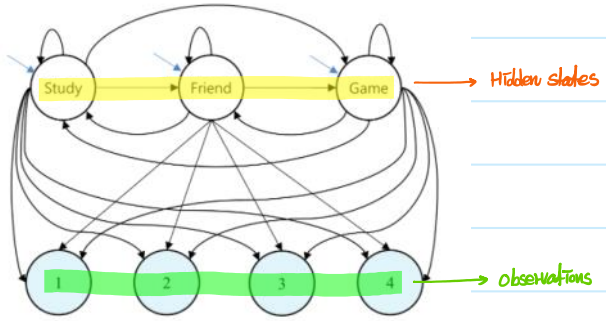
$\Leftrightarrow$ x_2 would be affected by the fact.

; Hence, the hidden variables are dependent for this case.

How does the HMM work ?

· In order for us to explain how the HMM work, let us take an example as below ;

- Let's say that the hidden state consists of studying, game, and friends.
- The observations are an expense history record such that $1 \sim 4$; where the higher, the more expense.
- In the end, we are expected to infer the past behavior based on the historical records in terms of expense.



· Now, let us assume that we already have the following probabilities ;

Initial probabilities	Study	Friend	Game
Initial	0.4	0.2	0.4
Study	0.4	0.3	0.3
Friend	0.7	0.1	0.2
Game	0.5	0.2	0.3

· This table includes transition probabilities !

e.g. Nov 23 : hang out with friends

→ Nov 24 : hang out with friends

⇒ 0.1 probability happened the situation

State \ Observation	1	2	3	4
Study	0.7	0.15	0.1	0.05
Friend	0.1	0.2	0.3	0.4
Game	0.5	0.2	0.1	0.2

· This table includes emission probability !

e.g. 0.7 probability to spend '1' expense when you are studying.

· It seems that these tables provide information regarding how people behave and expense habit according to the behaviors.

· Therefore, once the HMM (both the figure and tables) are given to us, we may be able to infer what we want to know.

- For example, let's say that the historical records are given as follows ;

11/20	11/21	11/22	11/23
1	1	4	2

- Using the HMM model with the values, we might be able to infer ; Probability that 1,1,4,2 is happen
  Behavior (e.g. Study, ...) that causes 1,1,4,2 pattern

- So, it sounds like that we can infer the behavior based on observations.

· However, we have to note that ;

- The tables are assumed to be given ; however, it's technically needed to estimate throughout the HMM process.

· Hence, when you model a HMM, you have to consider the following algorithms ;

1) Baum-Welch algorithm

: It is a type of EM algorithm. It's needed to estimate initial probability, transition probability, and emission probability.

2) Viterbi algorithm

: It is needed to estimate hidden states where it has high probability for the given observations. (i.e. 1,1,4,2 → states)

↳ Note that the probabilities (B, T, and I) are given.

3) Forward algorithm

: It is needed to estimate the probability where the observations are happen. The B,I,T probabilities are given.

· Let's walk through each algorithm !

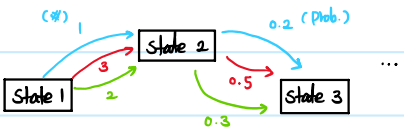
What is forward algorithm for a HMM ?

a) Motivation

· Let's say that we want to compute the likelihood of a particular observation sequence.

e.g. What is the probability of the sequence 3, 1, 3 (number of icecreams that Jason would eat) ?

· If a model is defined as Markov Model, we could easily compute the probability of 3, 1, 3 just by following the states labeled 3, 1, 3 and multiplying the probabilities along the arcs ;



Probability = 0.5 x 0.2 x ···

· However, if a model is defined as Hidden Markov Model, things are not so simple.

- This is because we don't know the hidden state sequence is. (e.g. weather)

· In terms of the HMM, we may be easily able to compute the likelihood if we already knew the weather.

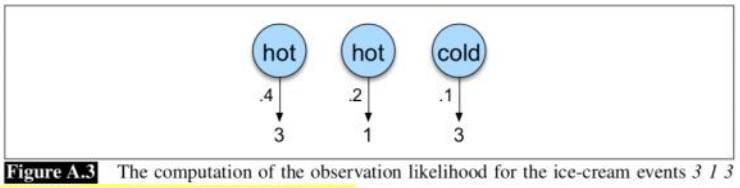


Figure A.3 The computation of the observation likelihood for the ice-cream events *3 1 3* given the hidden state sequence *hot hot cold*.

- Here, we know that each hidden state produces only a single observation in HMM ; thus, they have the same length T.

$$P(O|Q) = \prod_{i=1}^{T} P(O_i|Q_i) \quad ; \text{ where } O \text{ is observation and } Q \text{ is a hidden state sequence}$$

- For this case, we can say that ;

$$P(3,1,3 \mid hot, hot, cold) = P(3|hot) \times P(1|hot) \times P(3|cold)$$

- One more thing that we have to consider is that the current hidden state is also influenced by the previous state.

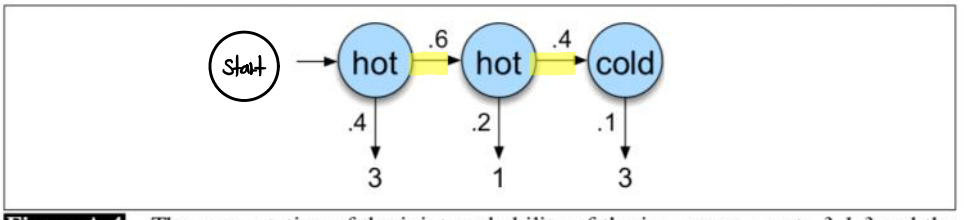


Figure A.4 The computation of the joint probability of the ice-cream events *3 1 3* and the hidden state sequence *hot hot cold*.

↳ $P(A|B) = \frac{P(A \cap B)}{P(B)} \Leftrightarrow P(A,B) = P(A|B) \cdot P(B)$

- Therefore, by considering the joint probability, $P(O,Q) = P(O|Q) \times P(Q) = \prod_{i=1}^{T} P(O_i|Q_i) \times \prod_{i=1}^{T} P(Q_i|Q_{i-1})$

$$P(3,1,3 \mid hot, hot, cold) = P(3|hot) \times P(1|hot) \times P(3|cold) \times P(hot|start) \times P(hot|hot) \times P(cold|hot)$$

· But of course, we don't actually know what the hidden state (i.e. weather) sequence was in reality.

· If we really want to calculate the likelihood of 3,1,3 events, we must consider all possible combinations ;

- Here, let us only think about two types of weather (hot and cold) and three states.

상태1	상태2	상태3
cold	cold	cold
cold	cold	hot
cold	hot	cold
hot	cold	cold
hot	hot	cold
cold	hot	hot
hot	cold	hot
hot	hot	hot

· Thus, the probability would be as follows :

$$P(3,1,3) = P(3,1,3 \mid \text{cold cold cold}) + P(3,1,3 \mid \text{cold cold hot}) + \cdots$$

· As can be seen, for real tasks, the total number of computations will be super high !

↳ This is the reason why the forward algorithm emerged.

b) Forward algorithm

· Instead of using such an extremely calculation volumes, we can use an efficient algorithm, called as forward algorithm.

· The forward algorithm is a kind of dynamic programming algorithm.

↳ It uses a table to store intermediate values and it will use them if necessary.

· Let us take an example.

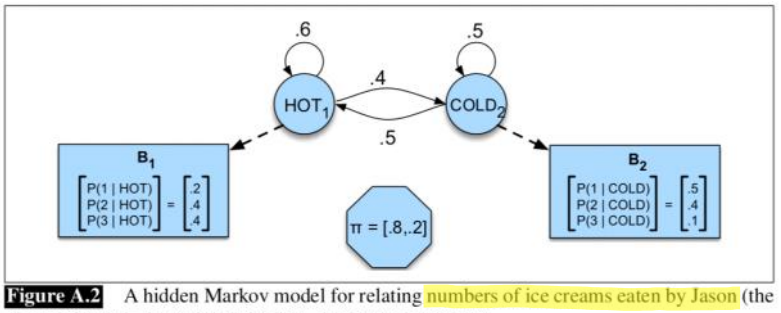
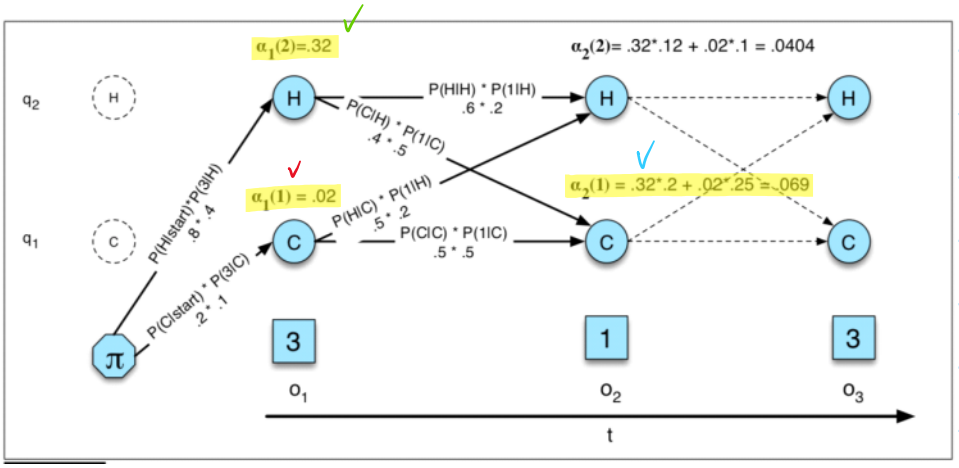


Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

· Now, let us apply the forward algorithm to the HMM.



- Let us think about the $d_1(1)$

: First of all, it's a probability where Jason would eat three icecreams when x_1 = cold.

$$d_1(1) = P(\text{cold} \mid \text{start}) \times P(3 \mid \text{cold})$$

- In a similar way, the probability $d_1(2)$

$$d_1(2) = P(\text{hot} \mid \text{start}) \times P(3 \mid \text{hot})$$

- Now, let's think about the $d_2(1)$ ✓

  1) Without the forward algorithm, we would need to calculate $P(3,1)$ for all possible combinations.

  2) With the forward algorithm, we may be able to use both $d_1(1)$ and $d_1(2)$ in order to reduce computational costs.

$$d_2(1) = d_1(1) \times P(\text{cold} \mid \text{cold}) \times P(1 \mid \text{cold}) + d_1(2) \times P(\text{cold} \mid \text{hot}) \times P(1 \mid \text{cold})$$

↓ A probability where Jason would sequentially eat three and one icecreams when t_2 = cold.

· Hence, the forward algorithm says that ;

- For a given state q_j at time t, the value $d_t(j)$ is computed as ;

$$d_t(j) = \sum_{i=1}^{N} d_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

- As can be seen, the equation is multiplied by three factors ;

$\alpha_{t-1}(i)$	the **previous forward path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j

· In summary,

- Given a Hidden Markov Model $\lambda = (A,B)$ (↳ something like emission probability) and an observation sequence O,

- What if we want to determine the likelihood $P(O|\lambda)$?

- Using the forward algorithm, we know that $P(O|\lambda) = d_t(q_t)$

What is Viterbi algorithm for a HMM ?

a) Introduction

· Given as input a HMM $\lambda = (A,B)$ and a sequence of observations $O = o_1, o_2, o_3, \cdots, o_T$,

- we may want to find the most probable sequence of hidden states $Q = q_1, q_2, q_3, \cdots, q_T$

· For example, in the icecream problem,

- Given a sequence of icecream observations 3,1,3 and a HMM, we want to find the best hidden weather sequence.

· In order for that, we might propose ;

" Let's calculate all possible hidden state sequence such as hot/hot/cold and then we could choose the hidden state sequence with maximum observation likelihood "

↳ It might work ; however, as always, it should be a problem when the domain becomes large.

· For this reason, the Viterbi algorithm is introduced.

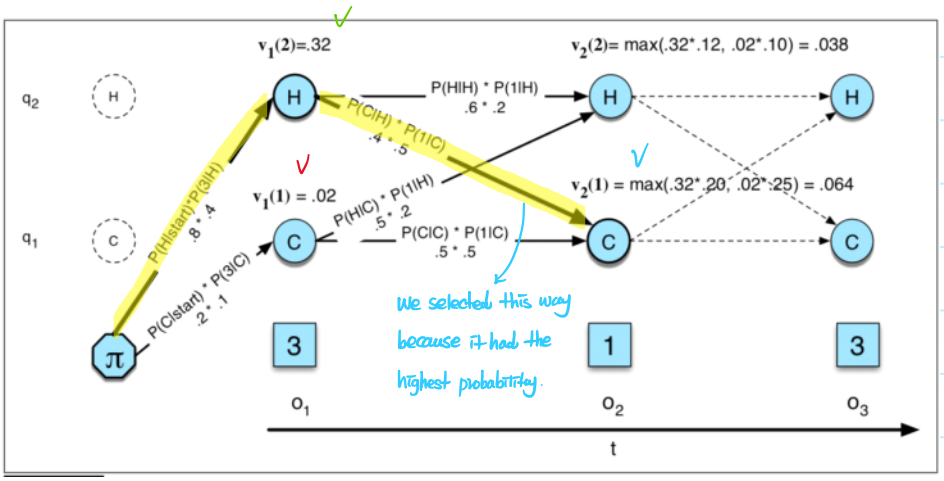
b) Viterbi algorithm

· The task of determining which sequence of variable is the underlying source of some sequence of observations is called as the decoding task.

· The most common decoding algorithms for HMM is the Viterbi algorithm.

↪ It's also a kind of dynamic programming.

· Let us take an example. In conclusion, the best hidden sequence is [Hot, cold] when 3, 1 is observed.



· Let us take a look three cases : $V_1(1)$, $V_1(2)$, and $V_2(1)$

$$
\begin{aligned}
v_1(1) &= \max\left[P(cold|start) \times P(3|cold)\right] \\
&= P(cold|start) \times P(3|cold) \\
v_1(2) &= \max\left[P(hot|start) \times P(3|hot)\right] \\
&= P(hot|start) \times P(3|hot) \\
v_2(1) &= \max\left[v_1(2) \times P(cold|hot) \times P(1|cold),\right. \\
&\qquad \left. v_1(1) \times P(cold|cold) \times P(1|cold)\right]
\end{aligned}
$$

↪ So, we are going to compare two cases to find the maximum probability.

· Therefore, the Viterbi algorithm says that :

- For a given state q_j at time t, the value $v_t(j)$ is computed as ;

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

· Note that

- The Viterbi algorithm is identical to the forward algorithm except that it takes the max over the previous path probabilities whereas the forward algorithm takes the sum.

What is Baum-Welch algorithm ?

· To begin with, for more details, I would take a look 'Stanford lecture note' and EM algorithm hand-written note by Junghyun.

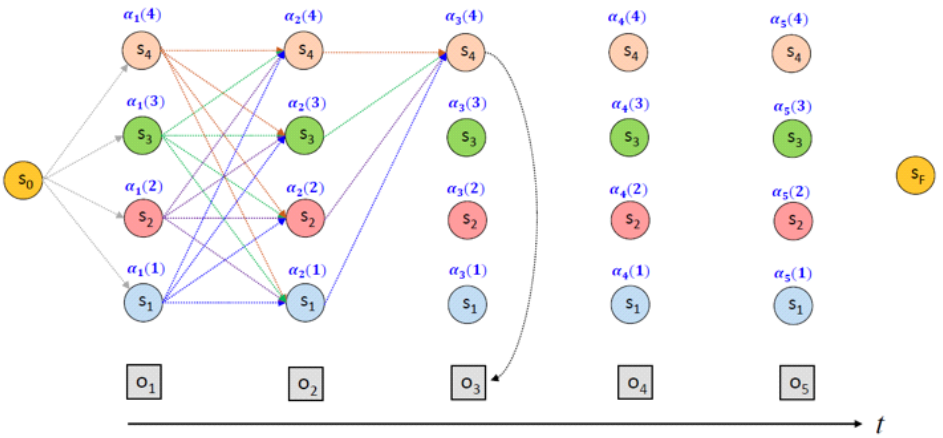
(In this hand-written note, I'll try to summarize it with high level)

- Basically, the purpose of Baum-Welch algorithm is to estimate the best values of θ based on the observations and hidden states sequences. (transition and emission probability)
- In other words,
  - The algorithm will let us train both the transition probability A and the emission probability B of the HMM.
- However, in real problems, it may be hard to estimate the probabilities at the same time.
  - For this reason, the algorithm would take an iterative way to estimate them.
    - ↳ This is why the algorithm is known as a special case of EM algorithm.
- Let us walk through the process:

1) Define the initial probability (π)

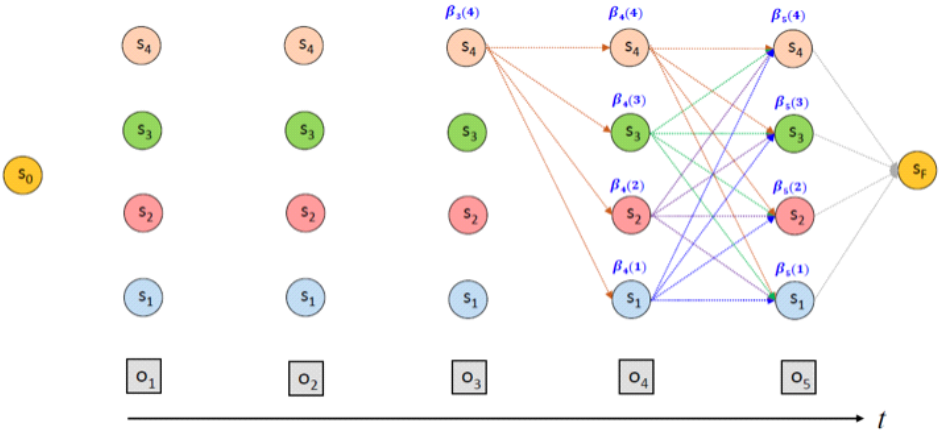
2) Initialize both transition and emission probabilities

3) E (Expectation) Step
   - Given the probabilities and observations,
   - Update / Calculate forward and backward probabilities (α and β)



" Forward "

$$\alpha_3(4) = \sum_{i=1}^{4} \alpha_2(i) \times a_{i4} \times b_4(o_3)$$



"Backward" $\beta_3(4) = \sum_{j=1}^{4} a_{4j} \times b_j(o_4) \times \beta_4(j)$

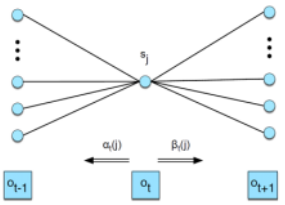
- Using both α and β, calculate γ and ξ

; where { γ = the expected state occupancy count
ξ = the expected state transition count

e.g.

$\gamma_t(j) = \frac{\alpha_t(j)\,\beta_t(j)}{P(O|\lambda)}$; where $P(O|\lambda) = \sum_{s=1}^{N} \alpha_t(s) \times \beta_t(s)$

This is because :



4) M (Maximization) step

- Use γ and ξ to recompute new A and B possibilities.

5) Repeat the process until convergence