



Maria Burlando

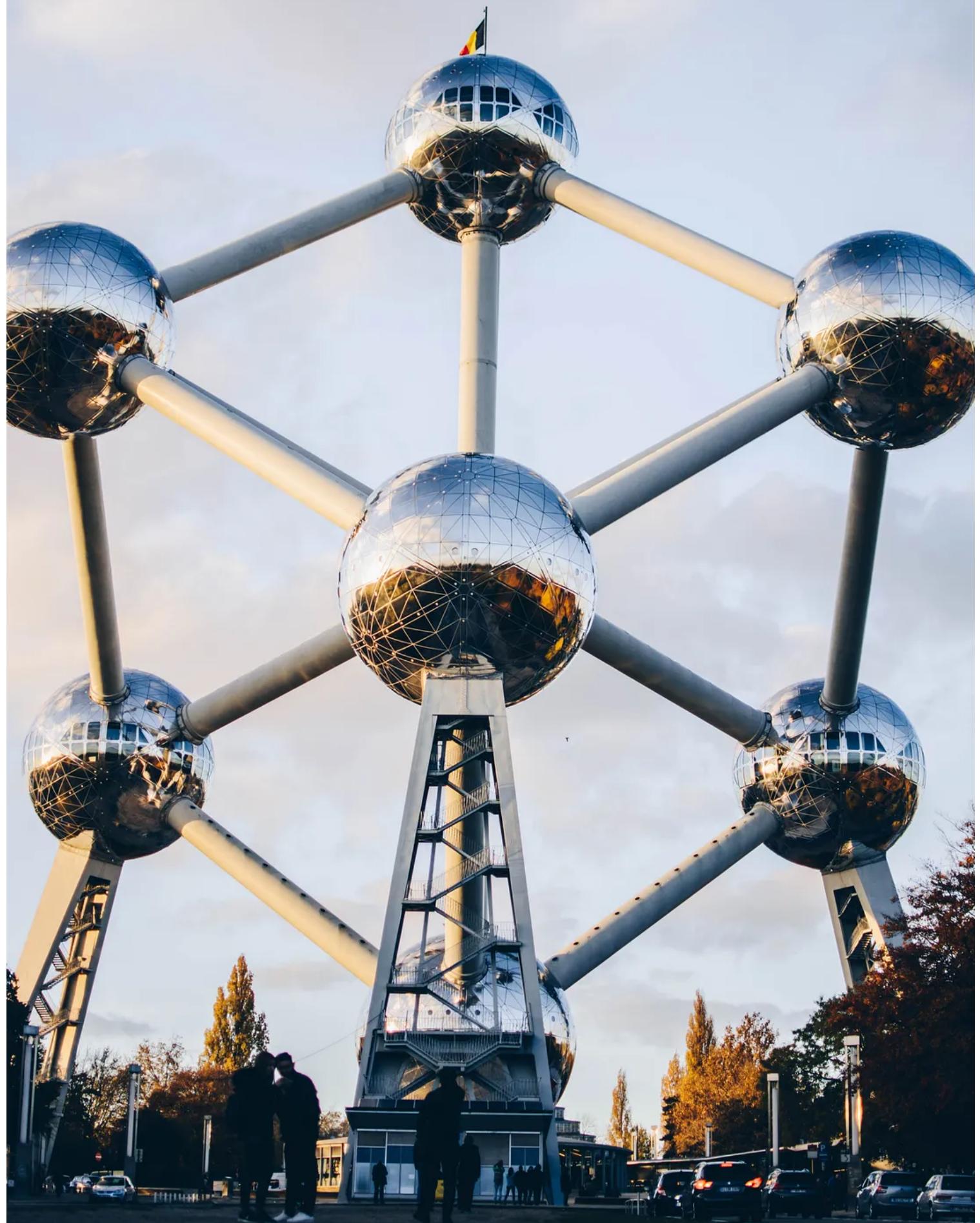
[Follow](#)

Dec 29, 2018 · 9 min read · [▶ Listen](#)

Save



## Hidden Markov Models — Part 1: the Likelihood Problem



**T**he traditional definition of HMM comes from Wikipedia's unfailing support when it comes to searching a new topic:

**Hidden Markov Model (HMM)** is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states.

And again, the definition for a Markov model:

*In probability theory, a Markov model is a stochastic model used to model randomly changing systems. It is assumed that future states depend only on the current state, not on the events that occurred before it (that is, it assumes the Markov property).*

Hidden Markov Models application include reinforcement learning and temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics.

Now that we've got the fancy descriptions out of the way, we're going to be looking at a rather simplified version of an HMM, so that our brains don't have to start frying.

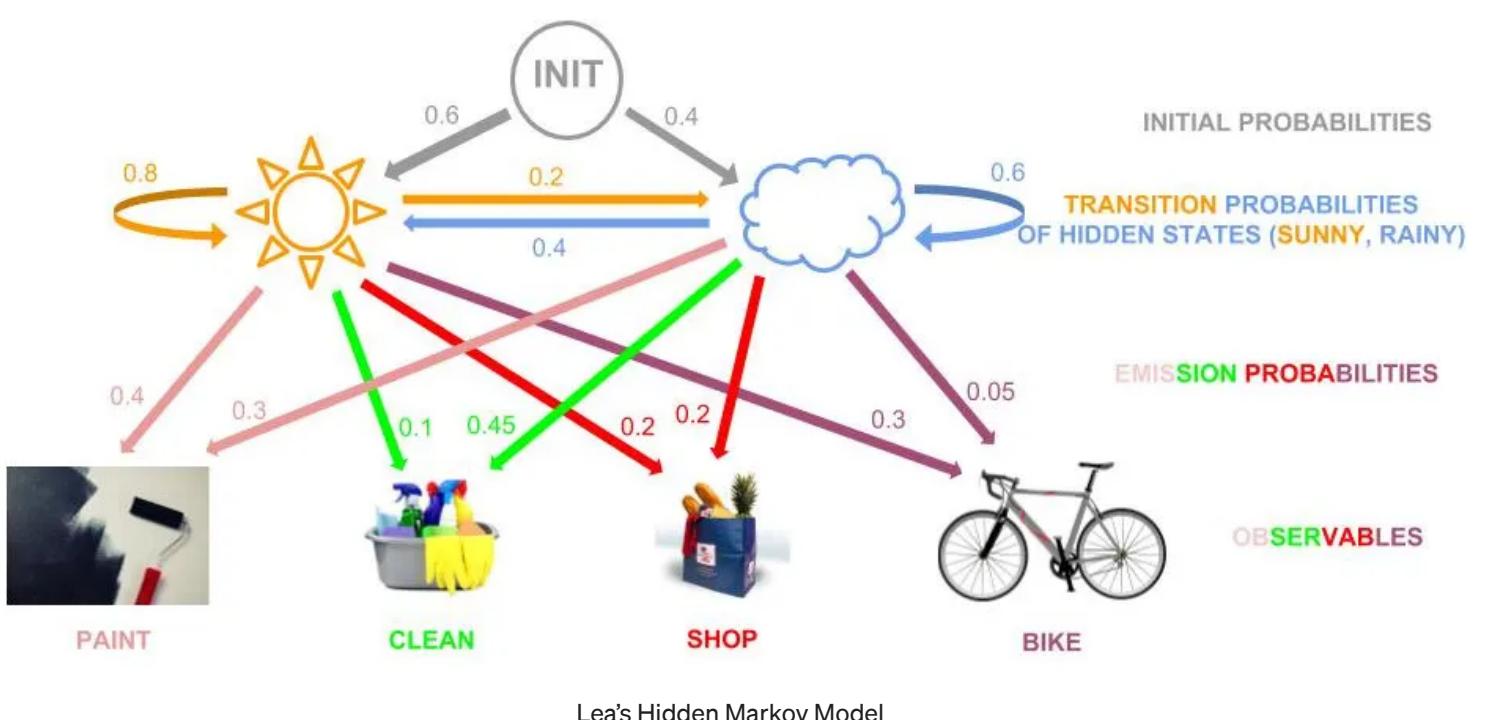
Let's take Lea; Lea is our imaginary friend and during the day she does either of these four things:

- Painting
- Cleaning the house
- Biking
- Shopping for groceries

Not much of a life, is it? But anyway, this is a list of what are called **observables**. Now, let's suppose that in four days Lea does the following: painting, cleaning, shopping, biking.

From this **observation sequence** we want to know whether the day has been sunny or rainy. These two are going to be our **hidden states**.

Here is a graph of our potential HMM:



To start off, a Hidden Markov Model consists of the following properties:

- **Hidden States  $S$ :** in the example above the hidden states are Sunny and Rainy, and they get grouped into a set  $S$ .
- **Observables  $O$ :** Paint, Clean, Shop and Bike. They get grouped into a set  $O$ .
- **Initial Probabilities  $\pi$ :** a matrix of the initial likelihood of the state at time  $t = 0$ . In this case the likelihood that it is Sunny on the first day is 0.6, while the likelihood that it is Rainy is 0.4.  
 $\pi = |0.6, 0.4|$   
*Note: every row of the following matrices must add up to 1 since they represent a probability.*
- **Transition Probabilities  $A$ :** a matrix that represents the probability of transitioning to another state given the current state. For example, if the current state is Sunny the probability that the day after is Sunny as well is 0.8, whereas the probability that the day after is Rainy is 0.2. Similarly if today is Rainy, the probability that tomorrow is Rainy as well is 0.6, while the probability that tomorrow is Sunny is 0.4.

			to	
from	Sunny	Rainy		
Sunny	0.8	0.2		
Rainy	0.4	0.6		

$$A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

Transition probabilities between states

- **Emission Probabilities  $B$ :** a matrix that represents the probability of seeing a specific observable given a hidden state. For example, the probability of Clean on a Sunny day is 0.1, whereas the probability of Clean on a Rainy day is 0.45.

					to	
from	PAINT	CLEAN	SHOP	BIKE		
Sunny	0.4	0.1	0.2	0.3		
Rainy	0.3	0.45	0.2	0.05		

$$B = \begin{vmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{vmatrix}$$

Emission probabilities

Under more mathematical terms, we would describe this model's properties as such:

$$S = \{ S_{sunny}, S_{rainy} \} \quad (\textit{Hidden States})$$

$$O = \{ O_{clean}, O_{bike}, O_{shop}, O_{paint} \} \quad (\textit{Observables})$$

$$\pi = | 0.6 \ 0.4 | \quad (\textit{Initial Probabilities})$$

$$A = \begin{vmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{vmatrix} \quad (\textit{Transition Probabilities})$$

$$B = \begin{vmatrix} 0.4 & 0.1 & 0.2 & 0.3 \\ 0.3 & 0.45 & 0.2 & 0.05 \end{vmatrix} \quad (\textit{Emission Probabilities})$$

This is all very nice, but immediately we are faced with three problems:

- The Likelihood problem
- The Decoding problem
- The Learning problem

In this first tutorial we're going to analyse the first problem, which asks the likelihood of a certain observation sequence deriving from the HMM model that we have initialized.

### **Problem 1 — Likelihood**

Let's take the initial example about Lea's activity in four days. The **observation sequence** is as follows: Paint, Clean, Shop and Bike.

$$O = \{ O_{paint}, O_{clean}, O_{shop}, O_{bike} \}$$

Observation sequence

So, what is the likelihood that this observation sequence  $O$  can derive from our HMM  $\lambda$ ?

$$P(O|\lambda) = ???$$

There are two methods with which we can calculate this: the Forward Algorithm and the Backward Algorithm.

#### **The Forward Algorithm**

The Forward algorithm comprises of three steps:

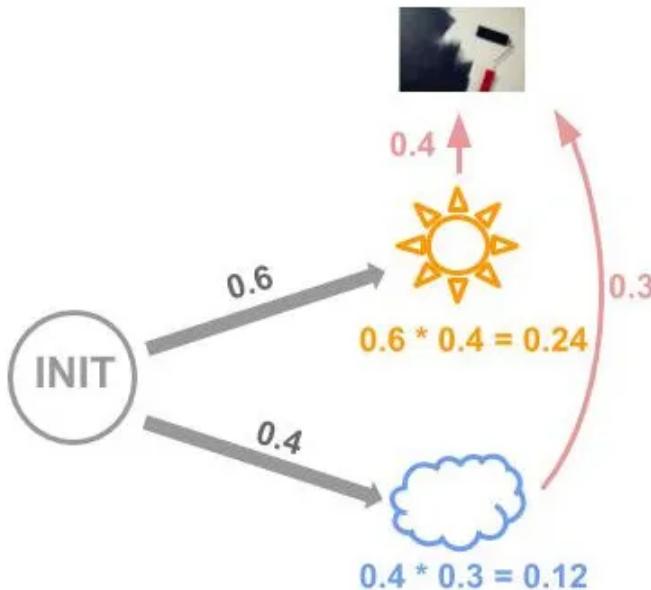
- Initialization
- Recursion
- Termination

## Initialization

$$\alpha_1(i) = \pi_i \cdot b_i(O_1)$$

Forward algorithm initialization equation

The above equation means that the first forward variable is calculated by multiplying the initial probability of state  $i$  by the emission probability  $b$  of that state given the observable  $O$  at time 1.



Initialization of Forward Algorithm

As can be seen the initial forward variable of the Sunny state is the initial probability of Sunny, 0.6, times the emission probability from Sunny to the observable Paint, 0.4. Hence, 0.24.

While the initial forward variable of the Rainy state is the initial probability of Rainy, 0.4, times the emission probability from Rainy to the observable Paint, 0.3. Hence, 0.12.

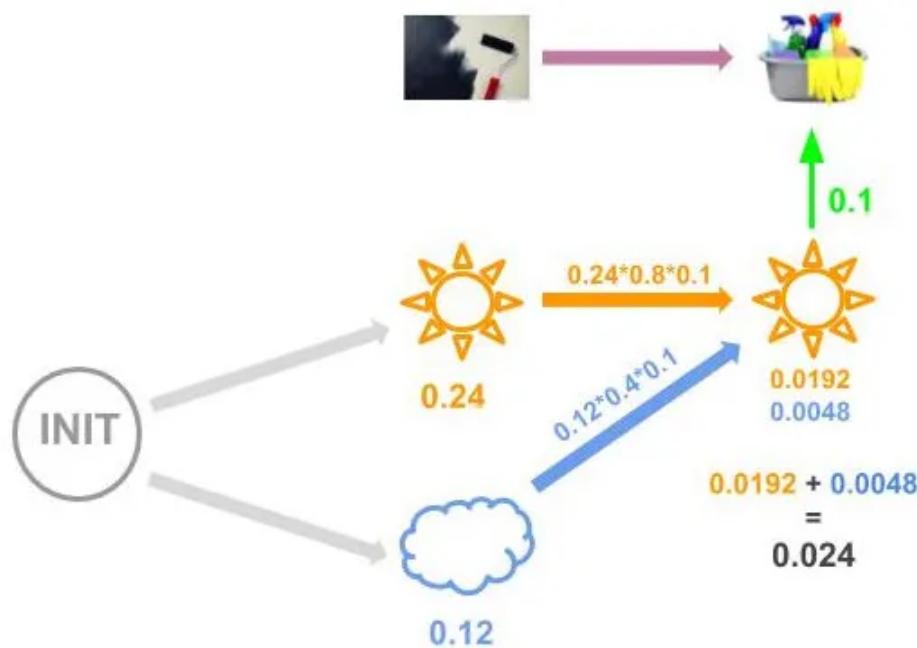
## Recursion

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

Forward algorithm recursion equation

For  $t = 1, 2, \dots, T-1$  we make use of the recursion equation which defines the forward variable of state  $j$  as the product of the previous forward variable of state  $i$ , multiplied by the transition probability  $a$  between the previous state  $i$  to state  $j$ , multiplied by the emission probability  $b$  from state  $j$  to the observable  $O$ .

Mind frying, I know, but let's have a look at the diagram below:



Recursion of Forward Algorithm 1

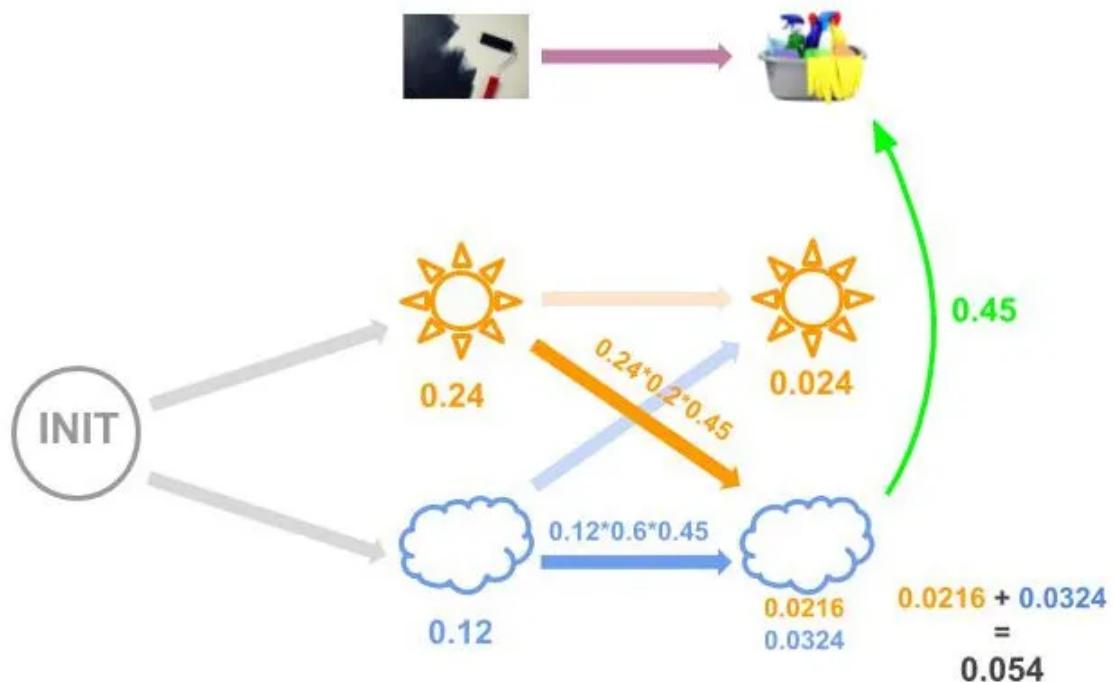
Here we are calculating the forward variable of state Sunny at time 2 by summing the results of two multiplications:

- The previous forward variable of the previous Sunny state, 0.24, times the transition probability from Sunny to Sunny, 0.8, times the emission probability from Sunny to Clean, 0.1.  
 $0.24 * 0.8 * 0.1 = 0.0192$
- The previous forward variable of the previous Rainy state, 0.12, times the transition probability from Rainy to Sunny, 0.4, times the emission probability from Sunny to Clean, 0.1.  
 $0.12 * 0.4 * 0.1 = 0.0048$

Then, according to the equation above, we sum these results and get our forward variable.

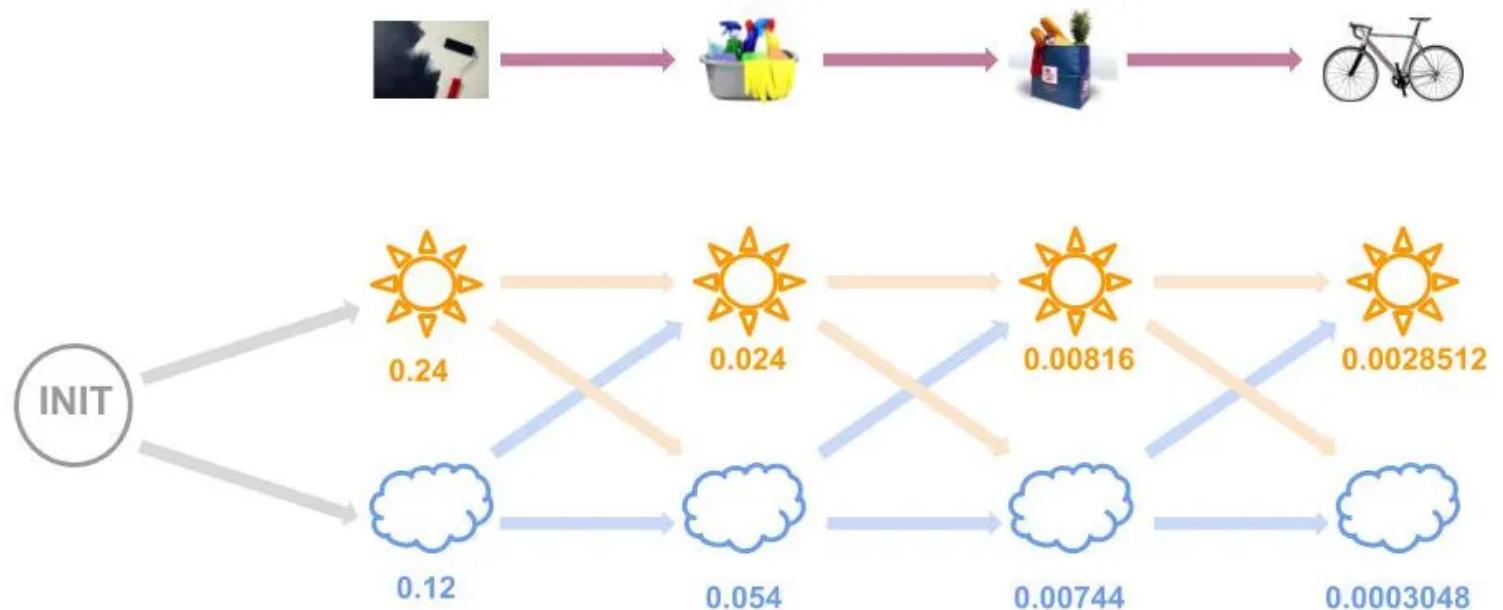
$$\alpha = 0.0192 + 0.0048 = 0.024$$

Similarly, for the next step we'll have a forward variable of 0.054 for the Rainy state:



Recursion of Forward Algorithm 2

And so on until we have all the forward variables:



Recursion of Forward Algorithm 3

## Termination

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

Termination

This final equation tells us that to find the probability of an observation sequence  $O$  deriving from an HMM model  $\lambda$ , we need to sum up all the forward variables at time  $T$ , i.e. all the variables of every state

at the end of the observation sequence. Hence, in our example above,

$$P(O|\lambda) = 0.0028512 + 0.0003048 = 0.003156$$

### The Backward Algorithm

Usually, to find the solution to the Likelihood problem we don't really require to know the Backward Algorithm. However, its explanation and resolution is a good litmus test to show that the Forward algorithm works proficiently, and moreover, by understanding it now, we can be ready for when the time will come to use it for the resolution of the third problem of Learning.

The Backward Algorithm is similar to the Forward algorithm, but like the name suggests it goes backward in time. There's again an Initialization, a Recursion and a Termination.

#### Initialization

$$\beta_T(i) = 1$$

Initialization

What this simple equation is telling us is that at time T (at the end of the observation sequence) the backward variables of every state is equal to 1. Simple as that.



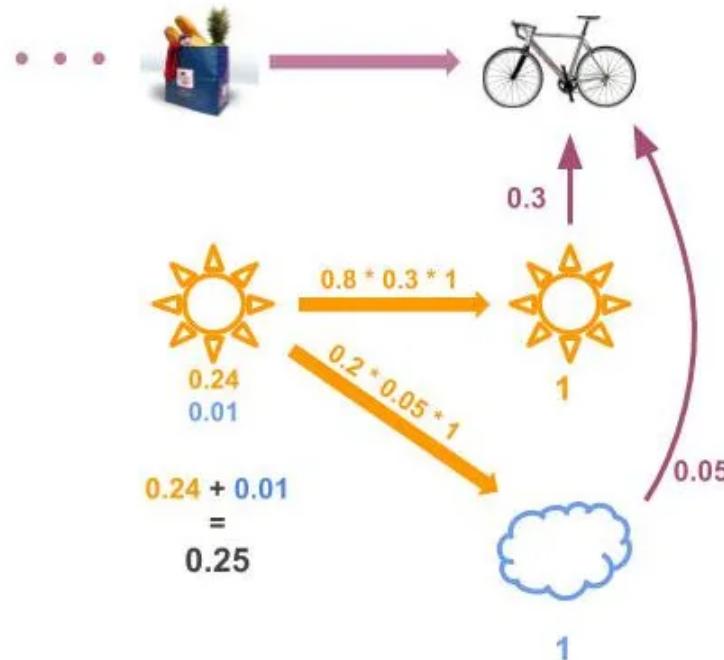
Initialization of Backward Algorithm

#### Recursion

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

The equation above tells us to sum up all the multiplications of the transition probability from the current State  $i$  to the next State  $j$ , times the emission probability of the observable  $O$  at time  $t+1$  from the next State  $j$ , times the backward variable of the next State  $j$  at time  $t+1$ .

It might not look extremely clear, but let's analyze the diagram below.



Recursion of Backward Algorithm 1

We're going back in time. At time  $T$  where the final observable in the observation sequence was Bike, the backward variables have been initialized as equal to 1.

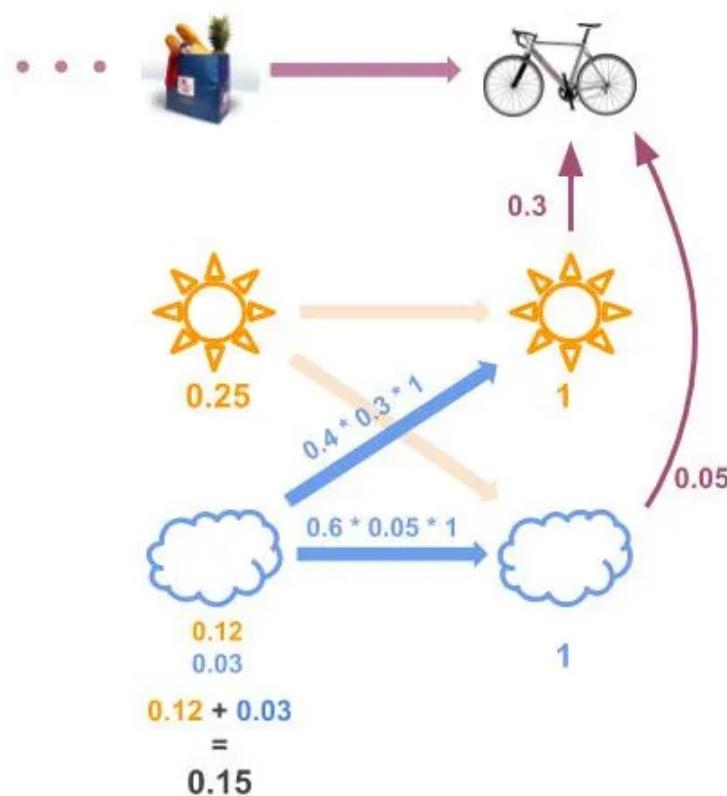
By going backwards we're going to calculate the backward variable of the previous Sunny state. This consists of summing up two multiplications:

- Transition probability from Sunny to Sunny, 0.8, times the emission probability of Bike deriving from the Sunny state at time  $t+1$ , 0.3, times the backward variable of the Sunny State at time  $t+1$ , 1.  
 $0.8 * 0.3 * 1 = 0.24$
- Transition probability from Sunny to Rainy, 0.2, times the emission probability of Bike deriving from the Rainy state at time  $t+1$ , 0.05, times the backward variable of the Sunny State at time  $t+1$ , 1.  
 $0.2 * 0.05 * 1 = 0.01$

Then, according to the equation above, we sum these results and get our backward variable.

$$\beta = 0.24 + 0.01 = 0.25$$

Similarly for the Rainy state:



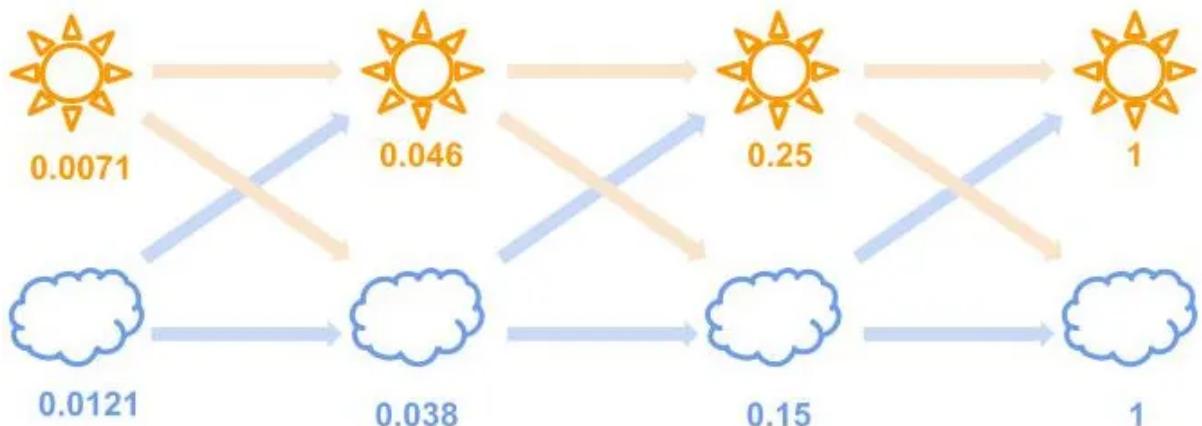
Recursion of Backward Algorithm 2

Like before, this is what we're going to sum up:

- Transition probability from Rainy to Sunny, 0.4, times the emission probability from the Sunny state to Bike at time  $t+1$ , 0.3, times the backward variable of the Sunny State at time  $t+1$ , 1.  
 $0.4 * 0.3 * 1 = 0.12$
- Transition probability from Rainy to Rainy, 0.6, times the emission probability from the Rainy state to Bike at time  $t+1$ , 0.05, times the backward variable of the Sunny State at time  $t+1$ , 1.  
 $0.6 * 0.05 * 1 = 0.03$

$$\beta = 0.12 + 0.03 = 0.15$$

And so on until we have all the backward variables:



Recursion of Backward Algorithm 3

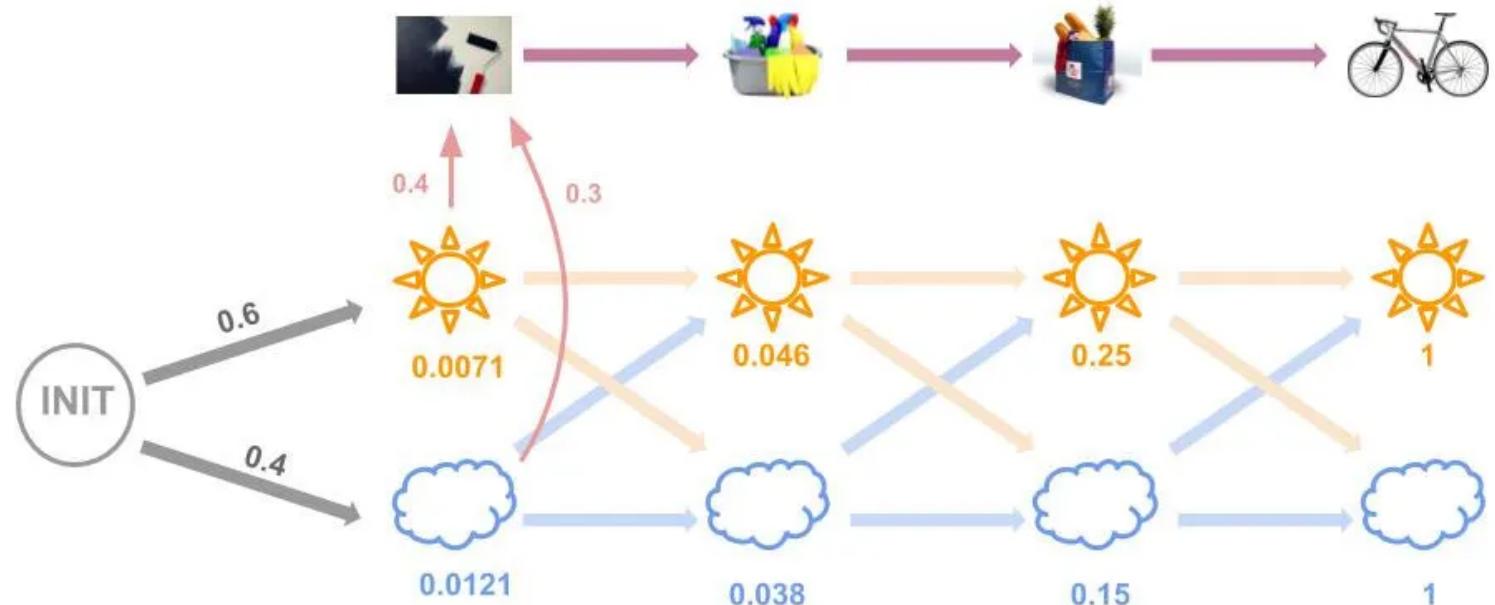
## Termination

$$P(O|\lambda) = \sum_{i=1}^N \pi_i \cdot b_i(O_1) \cdot \beta_1(i)$$

Termination

What this equation wants us to do is to sum the multiplications of the initial probability  $\pi$  of state  $i$ , times the emission probability  $b$  of the observable  $O$  at time  $t=1$  from state  $i$ , times the backward variable at time  $t=1$  of state  $i$ .

Let's clarify this in the diagram below:



Termination of Backward Algorithm



Search Medium



time  $t=1$  to the observable Paint, 0.4, times the backward variable of the Sunny state at time  $t=1$ , 0.0071.

$$0.6 * 0.4 * 0.0071 = 0.001704$$

- The initial probability  $\pi$  of the Rainy state, 0.4, times the emission probability from Rainy state at time  $t=1$  to the observable Paint, 0.3, times the backward variable of the Rainy state at time  $t=1$ , 0.0121.

$$0.4 * 0.3 * 0.0121 = 0.001452$$

$$P(O|\lambda) = 0.001704 + 0.001452 = 0.003156$$

And as we can see, the result is the same as that of the Forward Algorithm.

## Conclusion & What's next

To finalize what we've learned we can say that the probability that Lea has spent the last four days painting, cleaning, shopping and biking, is 0.003156 given the HMM model we have built for her.

### \*Bonus tip for developers\*

I know that these calculations are pretty hard to do by hand, and that is why I'm also offering you to play around the Lea example in JavaScript. I have created an [npm package called mary-markov](#) that solves all the problems of an HMM using the algorithms required.

You can download the npm package [here](#), and see how I wrote the JavaScript code [here](#).

Once you have installed it using `npm install --save mary-markov` you can write the following code in your `index.js` file and run it. This will solve you the Likelihood problem by calculating the Forward and Backward Algorithms.



454



1

\*\*\*\*\*

If you've managed to read up to this point, well done! This was certainly a hard one.

Unfortunately, it's only the beginning of the HMM saga. In the next article we're going to be looking at the Decoding problem of HMMs and how to solve it using the Viterbi Algorithm.

So, stick around, fellas!

Get the Medium app

