**IMPLEMENTATION OF HMM ALGORITHMS**

**(HIDDEN MARKOV MODELS)**

**1. INTRODUCTION**

- A Hidden Markov Model (HMM) is a statistical model that can be used to describe the evolution of observable events (Symbols) which they depend on hidden events (States).

- The HMM is based on augmenting the Markov chain. A Markov chain is a model

that tells us something about the probabilities of sequences of random variables,

states, each of which can take on values from some set. These sets can be words, or

tags, or symbols representing anything, like the weather.

- The HMM instantiates two simplifying assumptions:

a) **Markov Assumption:** the probability of a particular state depends

only on the previous state.

b) **Output Independence:** the probability of an output observation oi depends only on the state that produced the observation qi and not on any other states or any other observations.

- The HMM is composed of the following elements:

a) Q (sequence of N states)

b) O (sequence of T observations)

c) A (states transition probality matrix)

d) B (emission probabilities)

e) pi (initial states probabilities)

**2. IMPLEMENTATION**

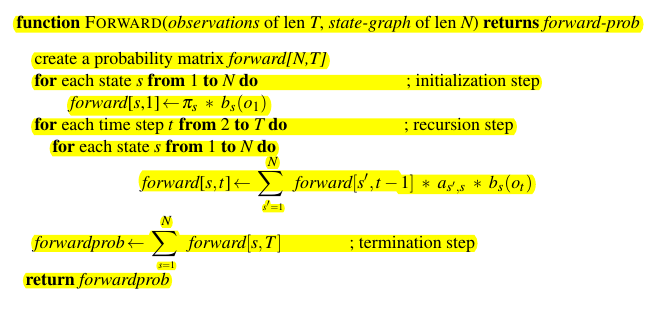
The HMM is characterized by three fundamental problems:

**2.1. Problem 1 (Likelihood):**

Given an HMM λ = (A, B) and an observation sequence O, determine the likelihood P(O|λ ).

**Likelihood**  is the probability of the observation sequence. The Forward algorithm computes the observation probability by summing over the probabilities of all possible hidden state paths that could generate the observation sequence.

This problem can be solved by using Forward algorithm , which calculate the probability of observation by adding the probabilities of all hidden states paths that could generate the observation sequence.

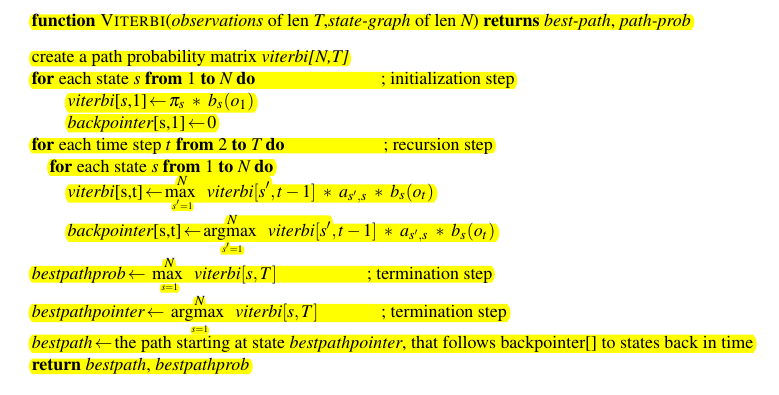


**2.2. Problem 2 (Decoding):**

Given an observation sequence O and an HMM λ = (A, B), discover the best hidden state sequence Q.

This problem can be solved by using Viterbi algorithm. It is identical to the Forward algorithm , except for the fact that it takes the max on the probabilities of the previous pah while the Forward algorithm takes the sum.

Note also, Viterbi algorithm has a component (Backpointer) that it holds trace of the path of hidden states that led to each state, and then at the end backtracing the best path to the beginning (the Viterbi backtrace).



**2.3. Problem 3 (Learning):**

Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

This problem can be solved using Forward-Backward algorithm (**The Baum-Welch algorithm**).

The Backward algorithm is the time-reversed version of the Forward Algorithm.

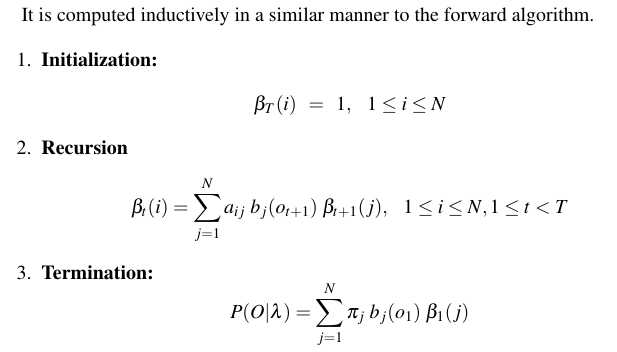
We will start with an estimate for the transition and observation

probabilities and then use these estimated probabilities to derive better and better

probabilities. And we’re going to do this by computing the Forward probability for

an observation and then dividing that probability mass among all the different paths that contributed to this forward probability.

We need to define a useful probability related to the Forward probability and called the Backward probability. The Backward probability β is the probability of seeing the observations from time t + 1 to the end :



The forward-backward algorithm has two steps: the expectation step, or E-step, and the maximization step, or M-step.

In the E-step, we compute the expected state occupancy count γ and the expected state transition count ξ from the earlier A and B probabilities. In the M-step, we use γ and ξ to recompute new A and B probabilities.

