

Hidden Markov Models

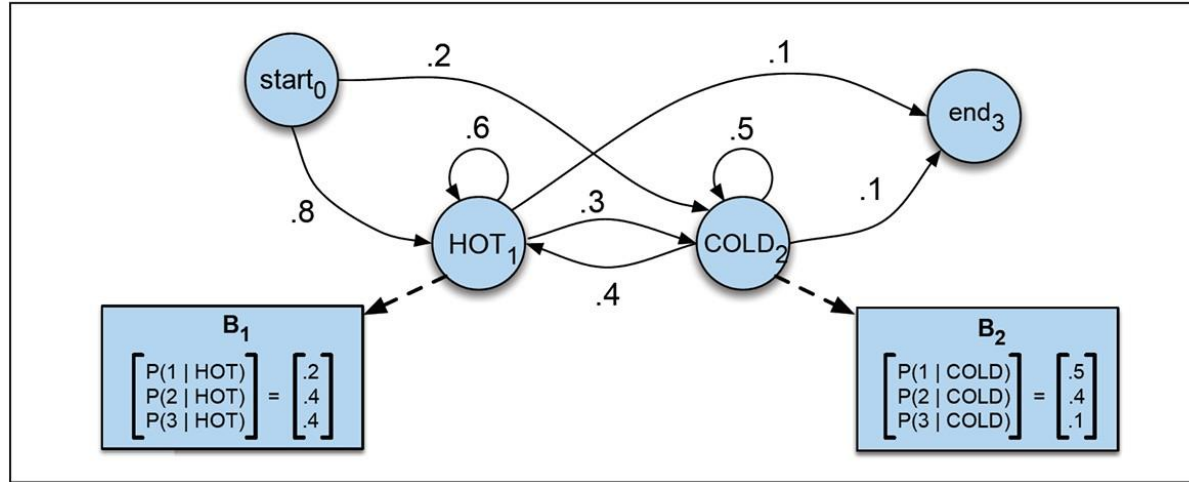
Assignment 3

Assignment: The Forward & Viterbi Algorithm

1. Implement the Forward Algorithm for the Hidden Markov Model (shown on the next slide) to compute the probability of the observation sequence 3 1 3.
2. Implement the Viterbi Algorithm to compute the most likely weather sequence for the observation sequence 3 1 3.

Use the file [hmm_template.py](#), it contains the incomplete functions [compute_forward](#) and [compute_viterbi](#).

Assignment : The Forward & Viterbi Algorithm



The Hidden Markov Model above shows the number of ice creams eaten by Jason (the observations) related to the weather (HOT or COLD, the hidden variables).

Visit <https://web.stanford.edu/~jurafsky/slp3/A.pdf> for further details.

Pseudocode for the Forward Algorithm

```
function FORWARD(observations of len  $T$ , state-graph of len  $N$ ) returns forward-prob

  create a probability matrix  $forward[N+2, T]$ 
  for each state  $s$  from 1 to  $N$  do                                ; initialization step
     $forward[s, 1] \leftarrow a_{0,s} * b_s(o_1)$ 
  for each time step  $t$  from 2 to  $T$  do                                ; recursion step
    for each state  $s$  from 1 to  $N$  do
       $forward[s, t] \leftarrow \sum_{s'=1}^N forward[s', t-1] * a_{s',s} * b_s(o_t)$ 

   $forward[q_F, T] \leftarrow \sum_{s=1}^N forward[s, T] * a_{s,q_F}$                 ; termination step
  return  $forward[q_F, T]$ 
```

Note that in the code, the transition matrix corresponds to [a](#), whereas the emissions matrix corresponds to [b](#).

Pseudocode for the Viterbi Algorithm

```
function VITERBI(observations of len  $T$ , state-graph of len  $N$ ) returns best-path

create a path probability matrix  $viterbi[N+2, T]$ 
for each state  $s$  from 1 to  $N$  do                                ; initialization step
     $viterbi[s, 1] \leftarrow a_{0,s} * b_s(o_1)$ 
     $backpointer[s, 1] \leftarrow 0$ 
for each time step  $t$  from 2 to  $T$  do                            ; recursion step
    for each state  $s$  from 1 to  $N$  do
         $viterbi[s, t] \leftarrow \max_{s'=1}^N viterbi[s', t-1] * a_{s',s} * b_s(o_t)$ 
         $backpointer[s, t] \leftarrow \operatorname{argmax}_{s'=1}^N viterbi[s', t-1] * a_{s',s}$ 
     $viterbi[q_F, T] \leftarrow \max_{s=1}^N viterbi[s, T] * a_{s,q_F}$         ; termination step
     $backpointer[q_F, T] \leftarrow \operatorname{argmax}_{s=1}^N viterbi[s, T] * a_{s,q_F}$  ; termination step
return the backtrace path by following backpointers to states back in
        time from  $backpointer[q_F, T]$ 
```

Note that in the code, the transition matrix corresponds to [a](#), whereas the emissions matrix corresponds to [b](#).