Viterbi Decoder

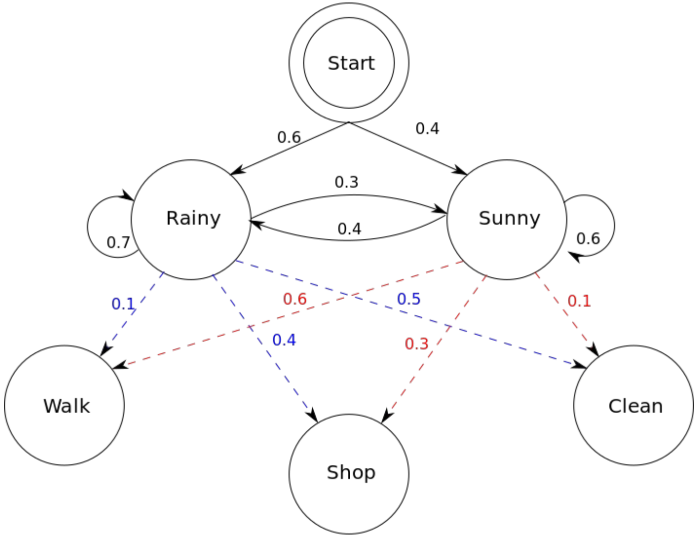
CS6230: CAD For VLSI

Hidden Markov Model

- **Markov Property:** Future state depends only on the current state and not the the events that occurred before it.

- **Hidden States:** Actual states of the system are not directly observable.

- **Observations:** Can only observe outputs generated by the hidden states



Viterbi Algorithm

- A dynamic programming algorithm that finds the **most likely sequence of hidden states** for a given sequence of observations made.
  Commonly used with Hidden Markov Models.

- **Viterbi path:** Resulting sequence of hidden states.

- Applications:
  - Decoding convolutional codes in digital communication
  - Parts of speech tagging

Mathematical Formulation

States: $q_0, q_1, q_2, ..q_N$ (q_0 = start state) N - no of states

Observations: $k_1, k_2, ...k_M$ M - no of observations

Transition probabilities: a_{ij} (probability of transitioning from state q_i to state q_j)

Emission probabilities: $b_j(o_t)$ (probability of observing `$o_t$` given the current state q_j)

Viterbi Path Probability: $v_t(j)$ (probability of the most likely path ending in state q_j at time t)

Input sequence of observations: $o_1, o_2, o_3, ..., o_T$ (o_t can be any of $k_1, k_2, ...k_M$)

(t > 1) $$\boxed{V_t(j) = \max_{(0<i<N+1)}(V_{t-1}(i) * a_{ij} * b_j(o_t))}$$

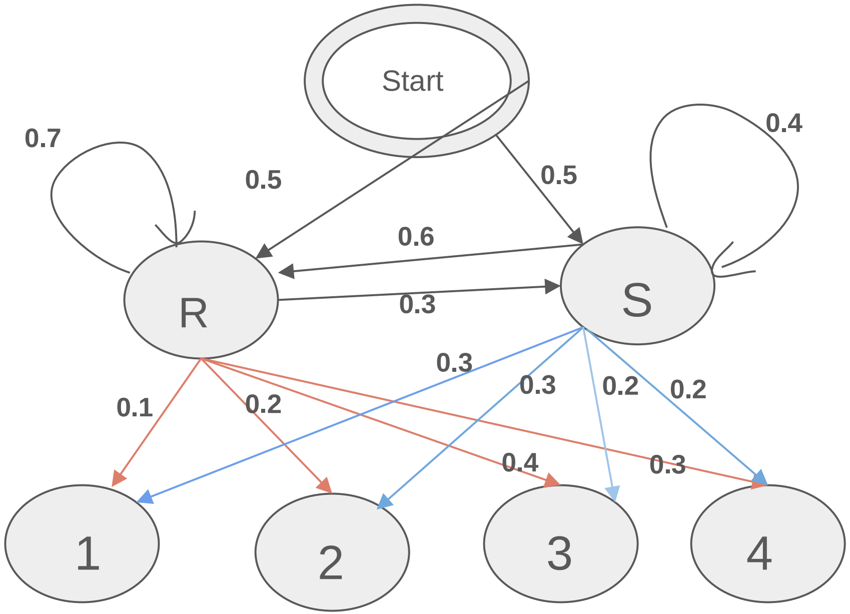
When t = 1, $V_1(j) = a_{q0,\,j} * b_j(o_t)$

Study material: https://web.stanford.edu/~jurafsky/slp3/A.pdf

Example

N = 2 (Rainy, Sunny)

M = 4 (Walk, Shop, Study, Clean)

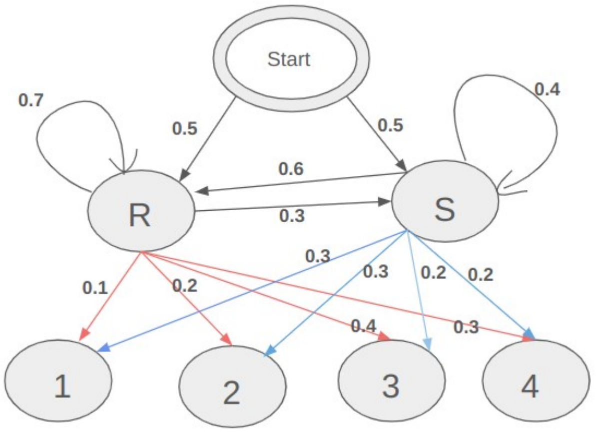


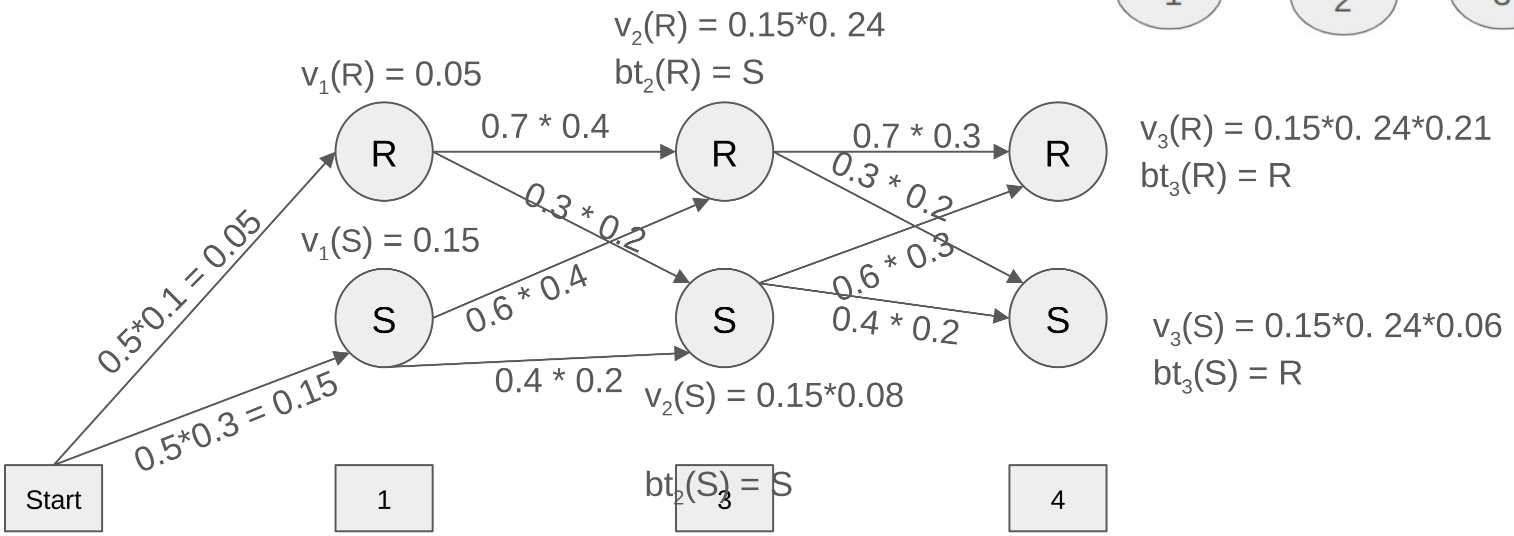
Observed for 3 consecutive days: 1,3,4 (W,S,C)   Most probable sequence of weather?

Observed for 3 consecutive days: 1,3,4 (W,S,C)

Most probable sequence of weather?

Ans: S,R,R (2,1,1) with prob = 0.15 * 0.24 * 0.21





Pseudocode

T - no of observations in input sequence
bt[t][j]: backtrace array which records prev state that led to the max probability ending at state j at time t
prob[]: current max path probability ending at each state
temp[]: temporary array to store updated values of probabilities at each time step

***Initialization(t = 1)*:** prob[j] = a[0][j] * b[j][o_1] for all j : 1 to N
bt[1][j] = q_0

Iterative step:

```
for each time step t from 2 to T do
    for each state j from 1 to N do
    {
        max_val = 0
        max_state = -1
        for i in 1 to N:
        {   val = prob[i] * a[i][j] * b[j][o_t]
            if val > max_val then:
                max_val = val;
                max_state = i;
        }
        temp[j] = max_val,   bt[t][j] = max_state
    }
    prob = temp; //copy temp array back to prob
```

Contd.

<u>Termination</u>**:**

path[]: final viterbi path of hidden states
vprob: final probability of mostly likely sequence

```
vprob =0;
for i in 1 to N:
{
    if prob[i] > vprob:
        vprob = prob[i] ;
        path[T] = i ;
}

for t in T-1 -> 1:
{
    path[t] = bt[t+1][path[t+1]];
}
```