Chapter IV. HMM

GUETTICHE Mourad

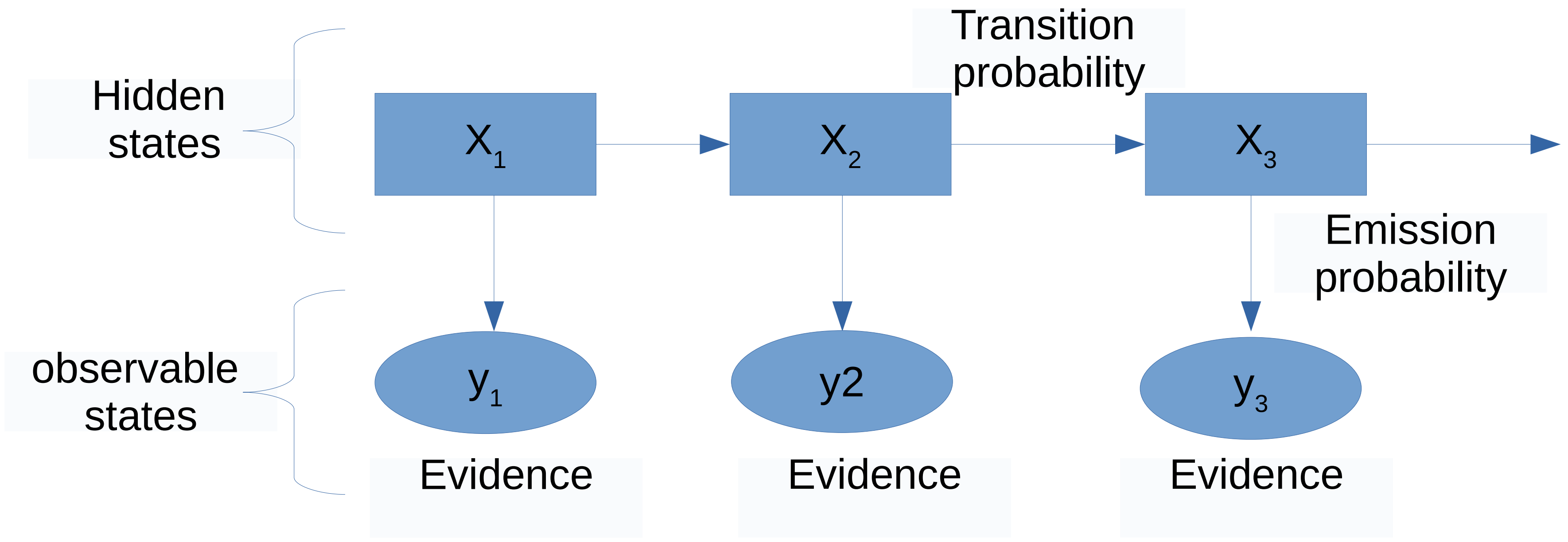
I. Markov chain

- A Markov chain describes a system whose state changes over time.
- The future of the system depends only to its present state, and not to the path by which the system go to this latter.
- A Markov chain is useful when we need to compute a probability for a sequence of observable events.

II. Hidden markov model

- In many cases, the events we are interested in are hidden.
- A hidden Markov model (HMM) allows us to talk about both observed events and hidden events that we think of as causal factors in our probabilistic model.

II.1 Definition



II. 1 Definition

An HMM is specified by the following components:

- $s_1, s_2, \ldots, s_N$: a set of N states.

- $A = [p_{ij}]_{n*n}$: a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j.

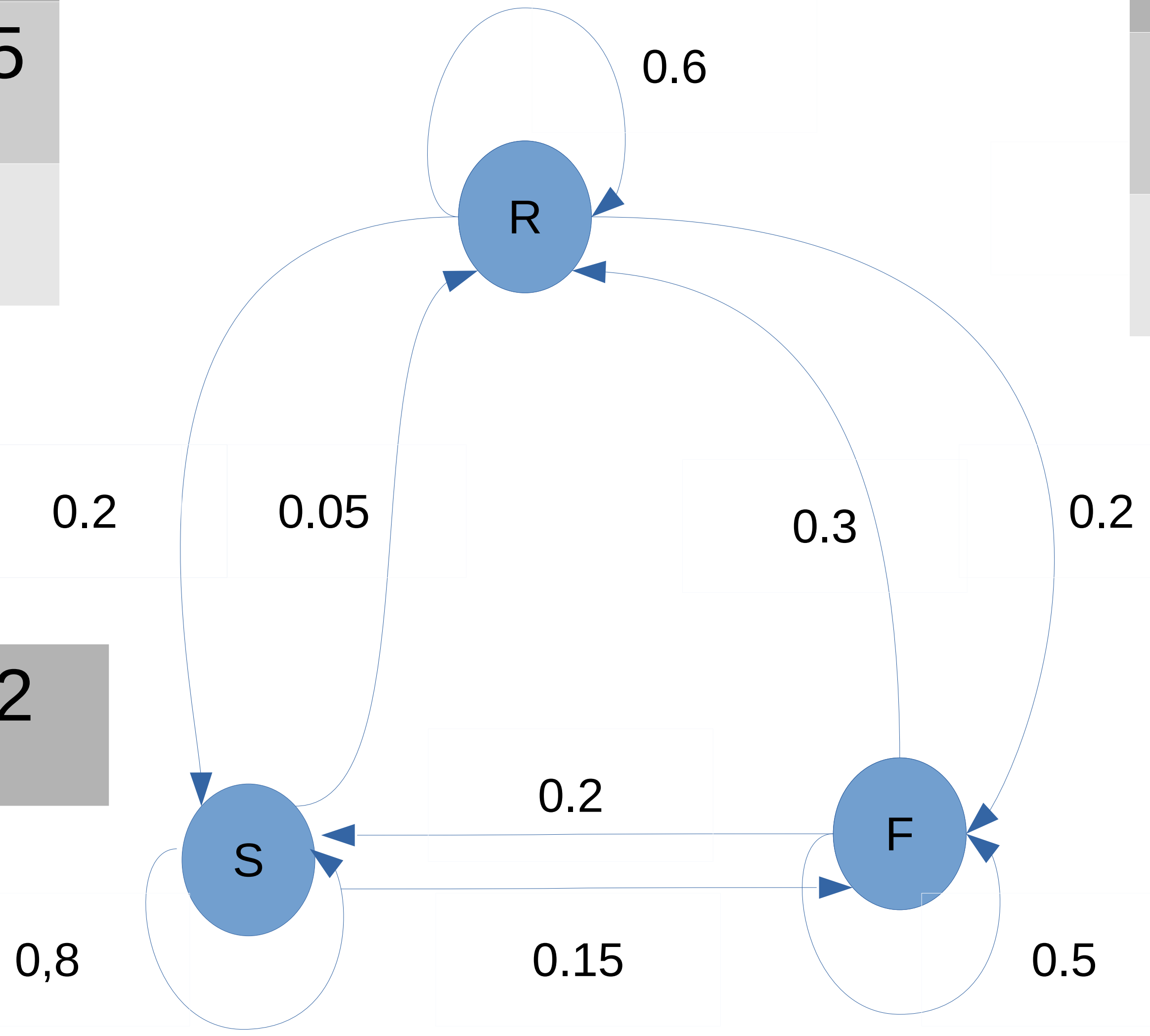
- $O = o_1 o_2 \ldots o_m$ a sequence of m observation symboles.

- $B = [o/s]_.$ emission probabilities, each expressing the probability of an observation o being generated from a state i.

-$\pi = \pi\ , \pi\ , \ldots, \pi_n$ an initial probability distribution over states.

0.6	0.2	0.2
0.05	0.8	0.15
0.3	0.2	0.5

0.3	0.7
0.95	0.05
0.5	0.5



0.1	0.7	0.2

III. Decoding

Example : given the following observable sequence.

Coat coat umbrella.

The different pssible sequence will bee.

Sunny sunny sunny, sunny rainy sunny, sunny foggy sunny,......

We have N^T possible cases with N the number of hidden states and T the size of the sequence.

$P(s_1,s_2,s_3/o_1,o_2,o_3)=P(s_1,s_2,s_3,o_1,o_2,o_3)/p(0_1,0_2,0_3)$.

$P(s_1,s_2,s_3,o_1,o_2,o_3)=P(o_1/s_1)*P(o_2/s_2)*P(o_3/s_3)*P(s_1)*p_{12}*p_{23}$.

$P(S,O)= \prod_{i=1}^{n} P(o_i/s_i)\,\pi(s_1)\,p_{i-1i}$

$S=Argmax_{s' \in ST}P(S',O)$.

III. 1 Viterbi Algorithm

Viterbi is a kind of dynamic programming that processes the observation sequence from left to right, filling out the cell.

Each cell $v_t(j)$, represents the probability that the HMM is in state j after seeing the first t observations and passing through the most probable state sequence S_1 , ..., S_{t-1} , given the automaton λ .

The value of each cell $v_t(j)$ is computed by recursively taking the most probable path that could lead us to this cell.

Formally, each cell expresses the probability

$v_t(j) = \max P(s_1 \ldots s_{t-1}, o_1, o_2 \ldots o_t, s_t = j)$

III. 1 Vitebi Algorithm

Note that we represent the most probable path by taking the maximum over all possible previous state sequences max $(s_1 \ldots s_{t-1})$

Like other dynamic programming algorithms, Viterbi fills each cell recursively.

For a given state S_j at time t, the value $v_t(j)$ is computed as :

$v_t(j) = \max^{n}_{i=1} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$.

III. 1 Viterbi Algorithm

Finally, we can give a formal definition of the Viterbi recursion as follows:

1. Initialization:

$v_1(j) = \pi_j b_j(o_1)$ $1 \le j \le N$

$bt_1(j) = \text{argmax } v_1(j)$ $1 \le j \le N$

2. Recursion

$v_t(j) = \max v_{t-1}(i)\, a_{ij}\, b\, j(o_t)$; $1 \le j \le N, 1 < t \le T$

$bt_t(j) = \text{argmax } v_t(j)$; $1 \le j \le N, 1 < t \le T$

3. Termination:

The best score: $P^* = \max v_T(i)$

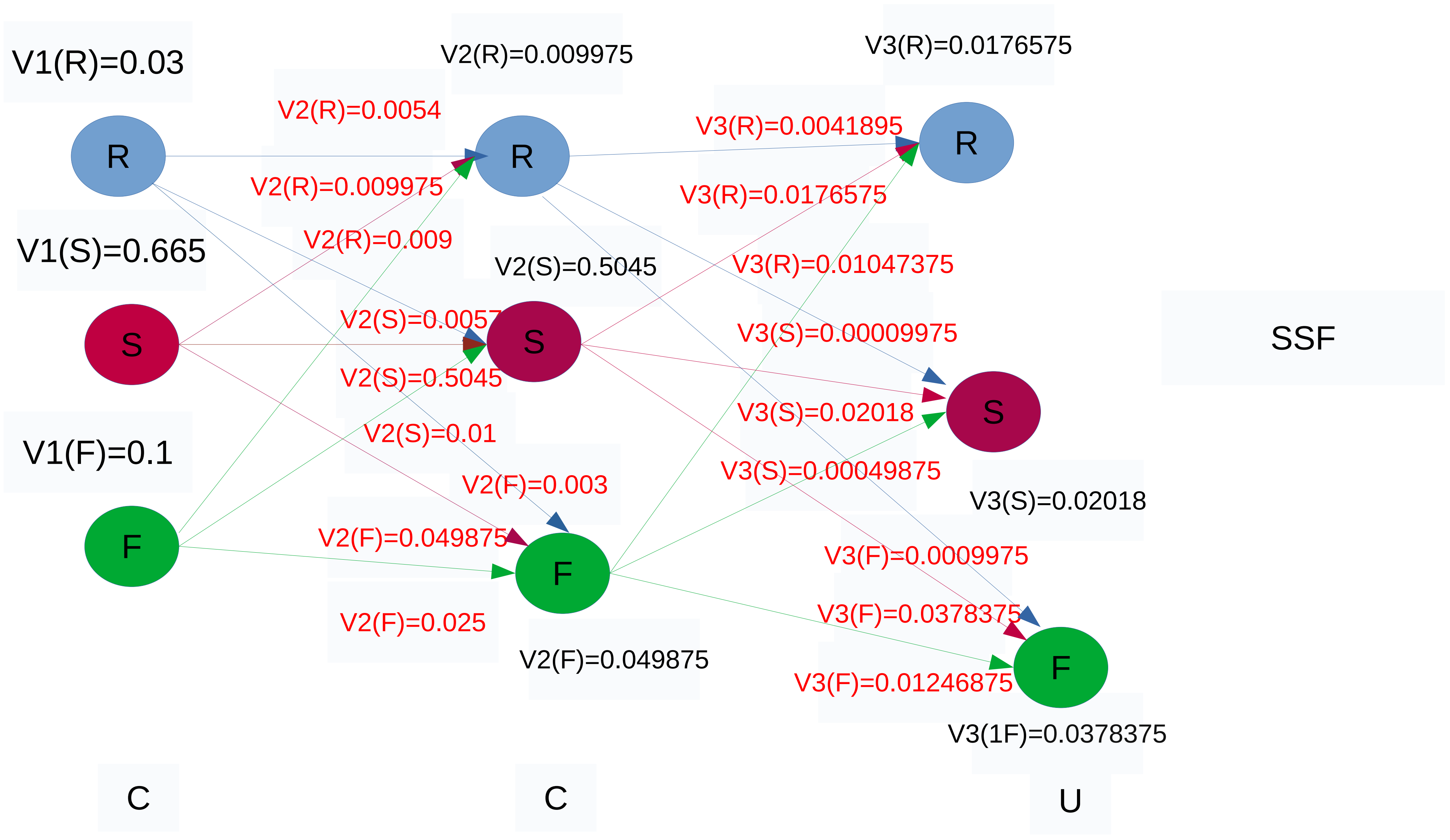
The start of backtrace: $q_T{}^* = \text{argmax } v_T(i)$

III. 2 Complexity

The complexity of veterbi algorithm is :

$O(N^2T)$.

Exampe : find the best sequence of the observation CCU using veterbi algorithm.


V1(R)=0.03
V1(S)=0.665
V1(F)=0.1
R
S
F
V2(R)=0.009975
V2(R)=0.0054
V2(R)=0.009975
V2(R)=0.009
V2(S)=0.5045
V2(S)=0.0057
V2(S)=0.5045
V2(S)=0.01
V2(F)=0.003
V2(F)=0.049875
V2(F)=0.025
V2(F)=0.049875
R
S
F
V3(R)=0.0176575
V3(R)=0.0041895
V3(R)=0.0176575
V3(R)=0.01047375
V3(S)=0.00009975
V3(S)=0.02018
V3(S)=0.00049875
V3(S)=0.02018
V3(F)=0.0009975
V3(F)=0.0378375
V3(F)=0.01246875
V3(1F)=0.0378375
R
S
F
SSF
C
C
U