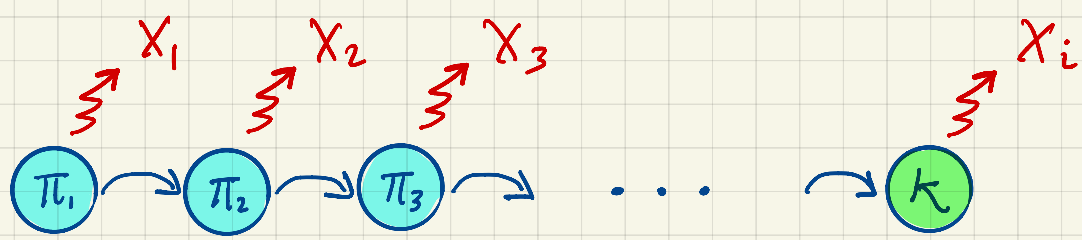
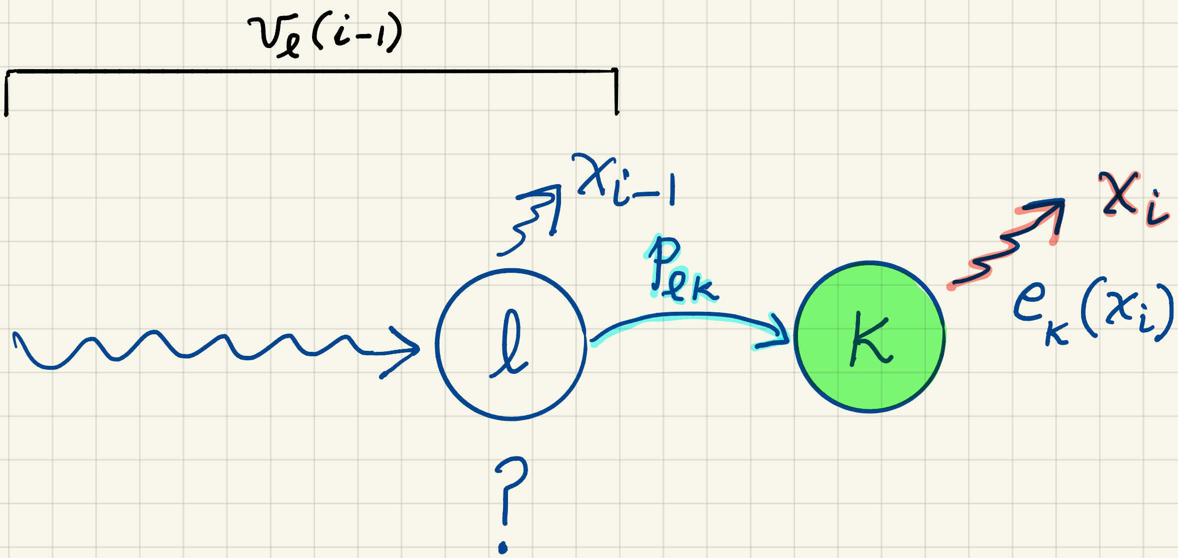
- Consider all possible paths
- Compute above probability for each path
- Identify the one with largest probability
- Correct, but NOT efficient.
- There are exponentially many path, e.g. 2^n for our example.

Solution: Viterbi algorithm.

Define $v_k(i)$ = prob. of <u>OPTIMAL</u> path $\pi_1 \dots \pi_i$ that generates $x_1 \dots x_i$ and ends in state k $(\pi_i = k)$





$$\boxed{v_k(i) = e_k(x_i) \max_{l} \left[v_l(i-1) \cdot p_{lk} \right]}$$

We can compute $v_k(i)$ if we have

$v_0(i-1), v_1(i-1), \ldots v_K(i-1)$ where K = # states

Finally, we need $\max_{k} v_k(n)$

Viterbi

$$V_0(0) = 1 \qquad V_k(0) = 0 \text{ for all } k \neq 0$$

$$V_k(i) = e_k(x_i) \max_{\ell} \left[V_\ell(i-1) \, P_{\ell k} \right]$$

$$\text{return } \max_k V_k(n)$$

Init

for $k = 0 \ldots K$

do $V[k,0] = 0$

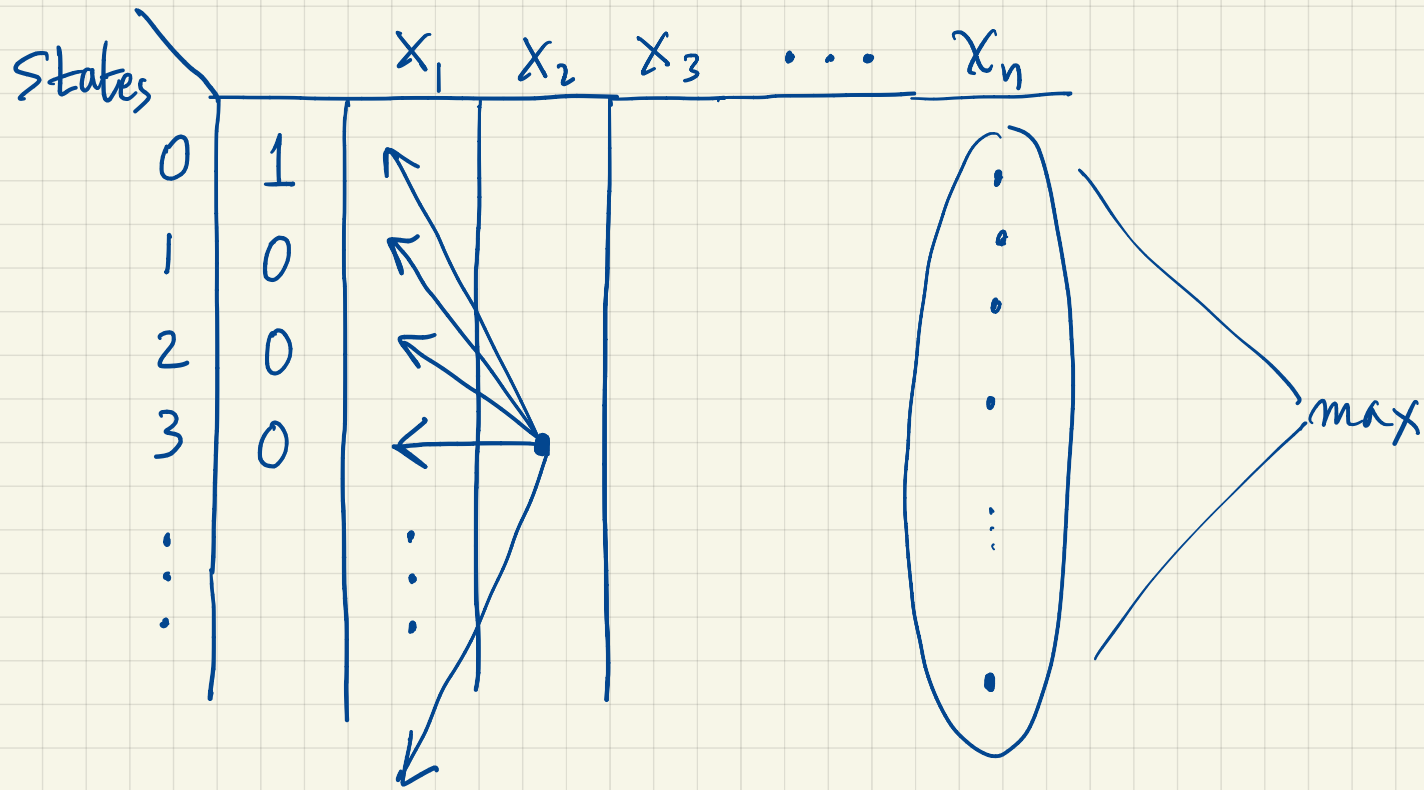
$V[0,0] = 1$

for $k = 0 \ldots K$

for $i = 1 \ldots n$

$$V[k,i] = e_k(x_i) \cdot \underbrace{\max_{\ell} \left[V[\ell, i-1] \cdot P[\ell,k] \right]}_{\text{for loop to find max}}$$

$v_0(0) = 1 \qquad v_k(0) = 0$ for $k \neq 0$

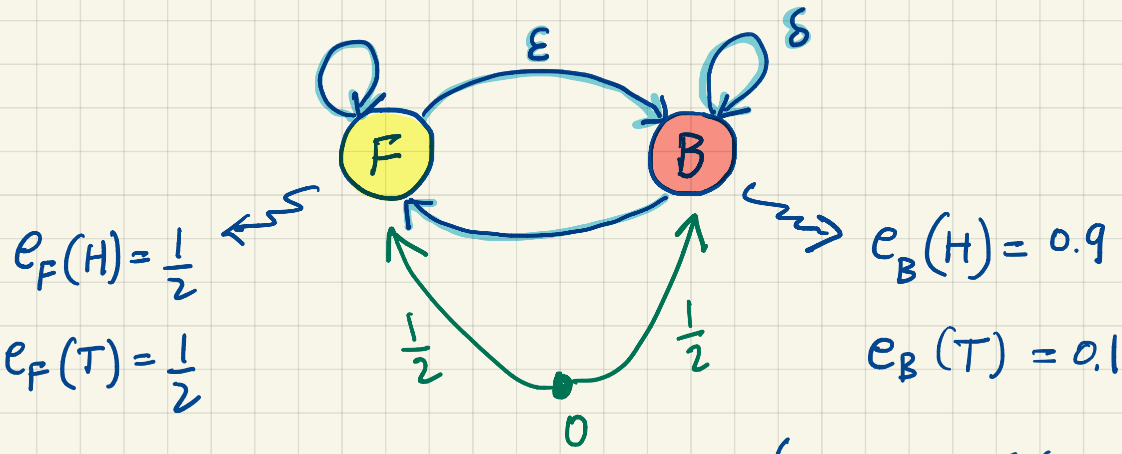


$v_k(i)$ needs the previous column to be computed.

Time needed proportional to nK^2, $K = \#$ states.

Example:

Assume $P_{0F} = P_{0B} = \frac{1}{2}$, $\varepsilon = \delta = 0.1$



		H	H
0	1	0	0
F	0	0.25	0.2025
B	0	0.45	0.0405

$$V_B(1) = \max \begin{cases} e_B(H)\, V_0(0)\, P_{0B} = 0.9 \times 1 \times \frac{1}{2} \\ e_B(H)\, V_F(0)\, P_{FB} = 0.9 \times 0 \times 0.1 \\ e_B(H)\, V_B(0)\, P_{BB} = 0.9 \times 0 \times 0.1 \end{cases}$$

$$V_F(2) = \max \begin{cases} e_F(H)\, V_0(1)\, P_{0F} = \frac{1}{2} \times 0 \times \frac{1}{2} \\ e_F(H)\, V_F(1)\, P_{FF} = \frac{1}{2} \times 0.25 \times 0.9 \\ e_F(H)\, V_B(1)\, P_{BF} = \frac{1}{2} \times 0.45 \times 0.9 \end{cases}$$

Practical Consideration

- Probabilities get too small ≈ 0 for large n.
- Problem with precision
- Use log space

$$V_k(i) = \log v_k(i)$$

$$E_k(b) = \log e_k(b)$$

$$P_{ek} = \log p_{ek}$$

Then

$$V_k(i) = E_k(b) + \max_{\ell} \left[V_\ell(i-1) + P_{\ell k} \right]$$

$$V_0(0) = 0 \qquad V_k(0) = -\infty \text{ for } k \neq 0$$

What if we want the most probable state for a given i

In general

$$P(\pi_i = k \mid x_1 \dots x_n) = \frac{P(x_1 \dots x_n, \pi_i = k)}{P(x_1 \dots x_n)}$$

First, let's consider $P(x_1 \dots x_n)$ and focus on $\boxed{P(x_1 \dots x_i, \pi_i = k)}$

Then $P(x_1 \dots x_n) = \sum_k \underbrace{P(x_1 \dots x_n, \pi_n = k)}_{f_k(n)}$

$f_k(i)$

Probability of generating $x_1 \dots x_i$ and ending in state k.

Similar recurrence to $v_k(i)$: $f_k(i) = e_k(x_i) \underbrace{\sum_l}_{} f_l(i-1) p_{lk}$

Replace max with $\sum$

Forward:

$$\boxed{\begin{aligned} &f_0(0) = 1 \qquad f_k(0) = 0 \ \text{ for all } k \neq 0 \\ &f_k(i) = e_k(x_i) \sum_{\ell} f_\ell(i-1)\, p_{\ell k} \\ &p(x_1 \dots x_n) = \sum_k f_k(n) \end{aligned}}$$

Now, let's look at $P(x_1 \dots x_n, \pi_i = k)$

$$P(\underbrace{x_1 \dots x_i}_{A} \overbrace{x_{i+1} \dots x_n}^{B}, \underbrace{\pi_i = k}_{A}) = P(\overbrace{x_1 \dots x_i, \pi_i = k}^{A})\, P(\overbrace{x_{i+1} \dots x_n}^{B} \mid \overbrace{x_1 \dots x_i, \pi_i = k}^{A})$$

$$= f_k(i) \cdot \underbrace{P(x_{i+1} \dots x_n \mid \pi_i = k)}_{b_k(i)}$$

Marko Property

$b_k(i)$ is the probability of generating $x_{i+1} \dots x_n$ given we are in state k at time i.

$$b_k(i) = \sum_{\ell} p_{k\ell}\, e_\ell(x_{i+1})\, b_\ell(i+1)$$

"backwards"

we need $b_\ell(i+1)$ for all ℓ to compute $b_k(i)$

Backward:

$$\boxed{\begin{array}{l} b_k(n) = 1 \quad \text{for all } k \\ b_k(i) = \sum_{\ell} p_{k\ell}\, e_\ell(x_{i+1})\, b_\ell(i+1) \\ \sum_{\ell} p_{0\ell}\, e_\ell(x_1)\, b_\ell(1) \end{array}}$$

$i = n-1 \ldots 1$

Note $b_0(0) = \sum_{\ell} p_{0\ell}\, e_\ell(x_1)\, b_\ell(1) = P(x_1 \ldots x_n)$

Finally $$P(\pi_i = k \mid x_1 \ldots x_n) = \frac{f_k(i)\, b_k(i)}{\sum_k f_k(n) = b_0(0)}$$