Genome 540 Class 18

Chengxiang Qiu

$$\alpha_1(1) = \pi_1 \times b_1(A) = 0.8 \times 0.4 = 0.32$$

Forward Algorithm

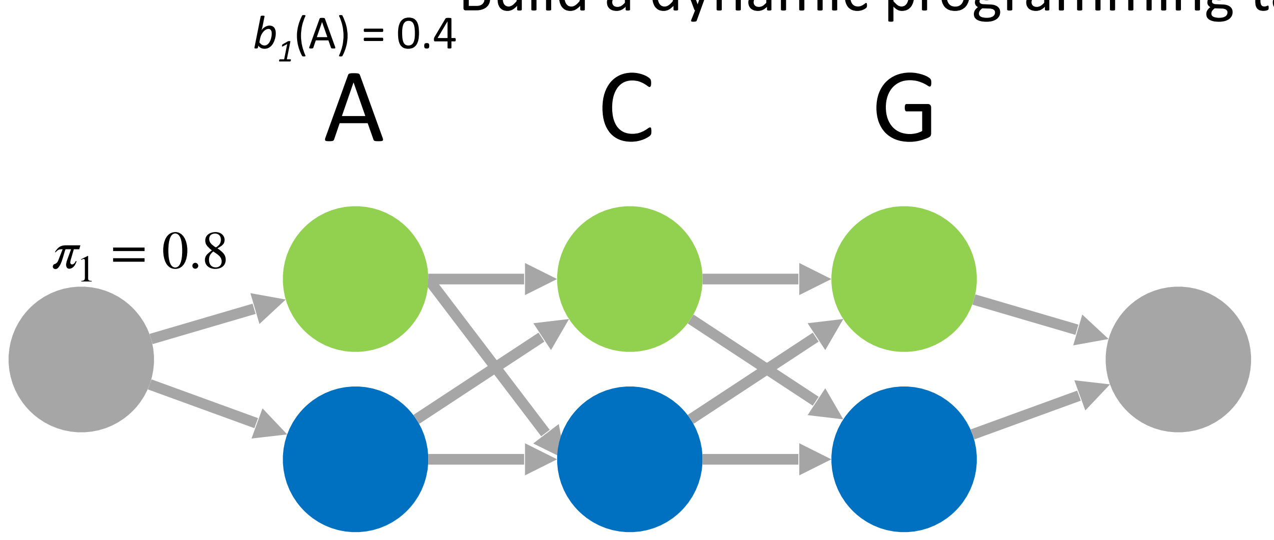
1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations



$\alpha_1(1)$

	A	C	G
State 1	0.32		
State 2			

$$\alpha_1(2) = \pi_2 \times b_2(A) = 0.2 \times 0.1 = 0.02$$

Forward Algorithm

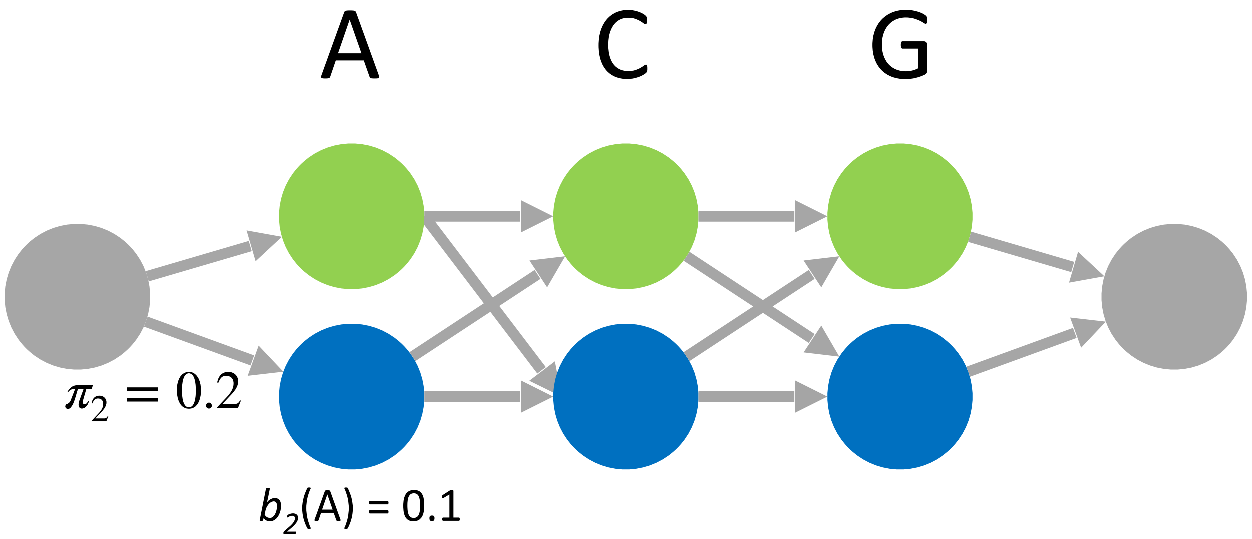
1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations



$\alpha_1(1)$

	A	C	G
State 1	0.32		
State 2	0.02		

$\alpha_1(2)$

$$\alpha_2(1) = [\alpha_1(1) \times a_{11} + \alpha_1(2) \times a_{21}] \times b_1(C)$$
$$= [0.32 \times 0.6 + 0.02 \times 0.5] \times 0.2$$
$$= 0.0404$$

Forward Algorithm

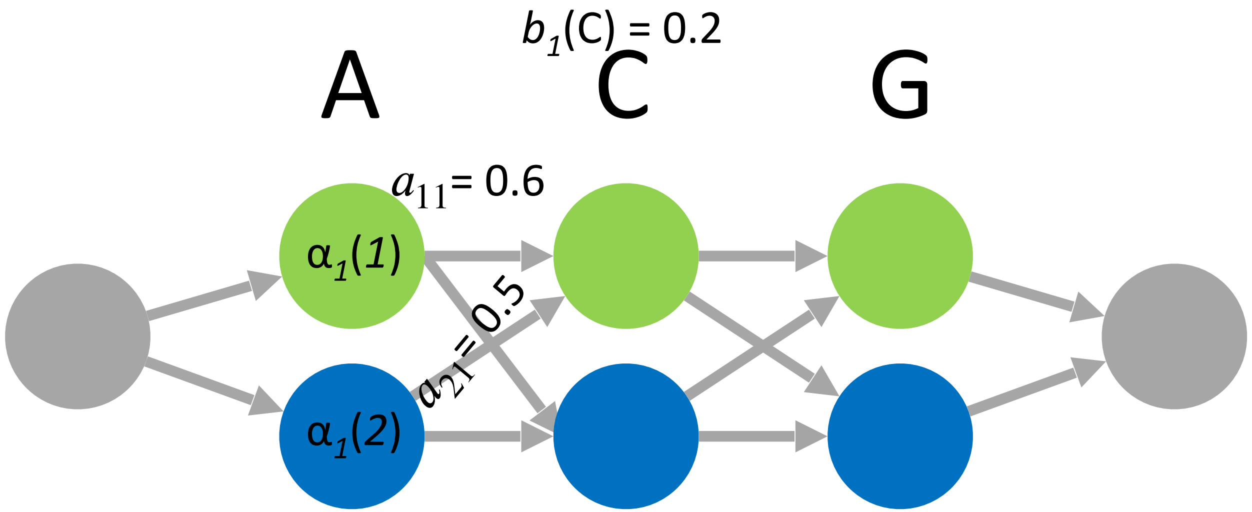
1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \leq t \leq T-1, 1 \leq j \leq N.$$

Build a dynamic programming table for these calculations



α$_1$(1) α$_2$(1)

	A	C	G
State 1	0.32	0.0404	
State 2	0.02		

α$_1$(2)

Forward Algorithm

$$\alpha_2(2) = [\alpha_1(1) \times a_{12} + \alpha_1(2) \times a_{22}] \times b_2(C)$$
$$= [0.32 \times 0.4 + 0.02 \times 0.5] \times 0.5$$
$$= 0.069$$

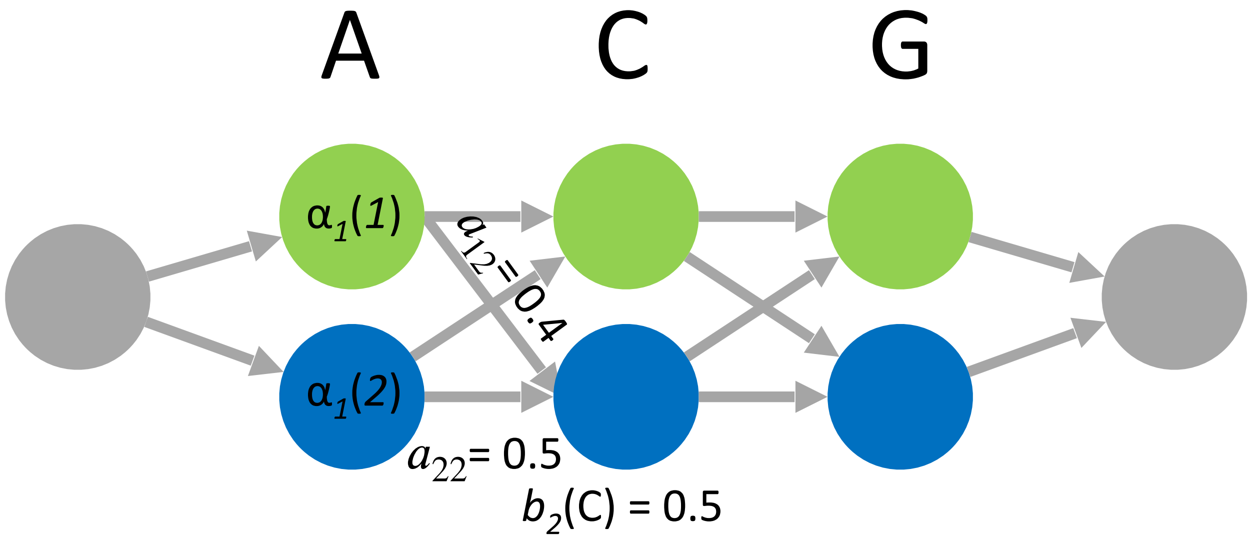
1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \leq t \leq T-1, 1 \leq j \leq N.$$

Build a dynamic programming table for these calculations



	$\alpha_1(1)$	$\alpha_2(1)$	$\alpha_3(1)$
	A	C	G
State 1	0.32	0.0404	
State 2	0.02	0.069	
	$\alpha_1(2)$	$\alpha_2(2)$	$\alpha_3(2)$

Backward Algorithm

$$\beta_2(1) = \beta_3(1) \times a_{11} \times b_1(G) + \beta_3(2) \times a_{12} \times b_2(G)$$
$$= 1 \times 0.6 \times 0.4 + 1 \times 0.4 \times 0.1$$
$$= 0.28$$

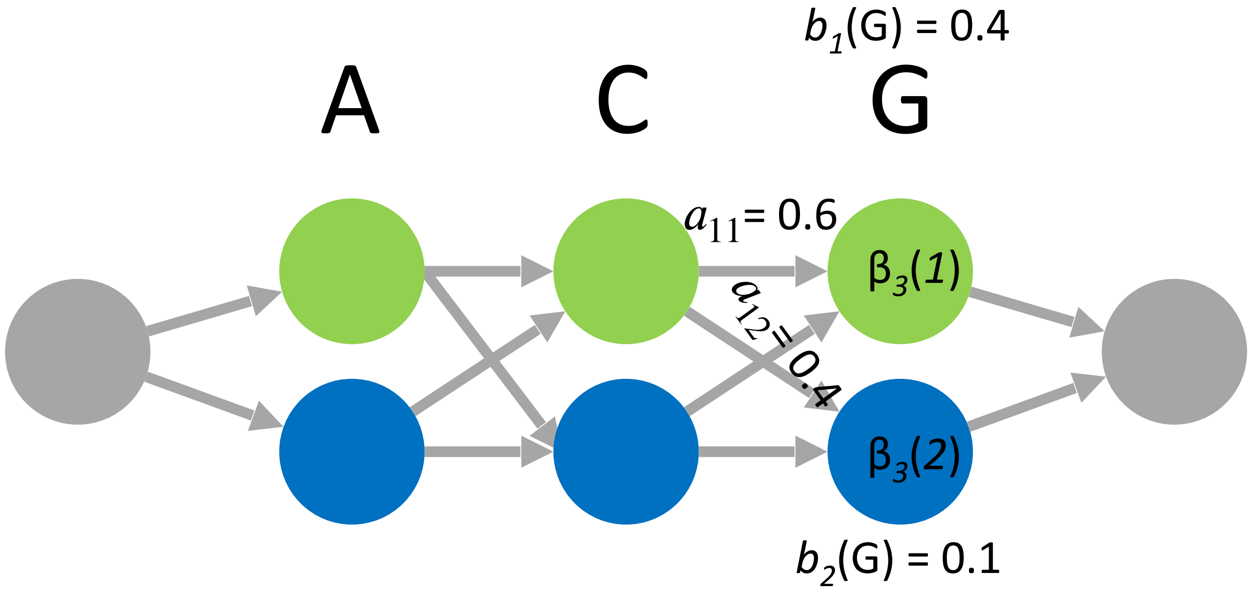
1. Initialization:

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N$$

2. Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \qquad 1 \leq t \leq T-1, 1 \leq j \leq N.$$

Build a dynamic programming table for these calculations



	$\beta_1(1)$	$\beta_2(1)$	$\beta_3(1)$
	A	**C**	**G**
State 1		0.28	1
State 2			1
	$\beta_1(2)$	$\beta_2(2)$	$\beta_3(2)$

Backward Algorithm

$$\beta_2(2) = \beta_3(1) \times a_{21} \times b_1(G) + \beta_3(2) \times a_{22} \times b_2(G)$$
$$= 1 \times 0.5 \times 0.4 + 1 \times 0.5 \times 0.1$$
$$= 0.25$$

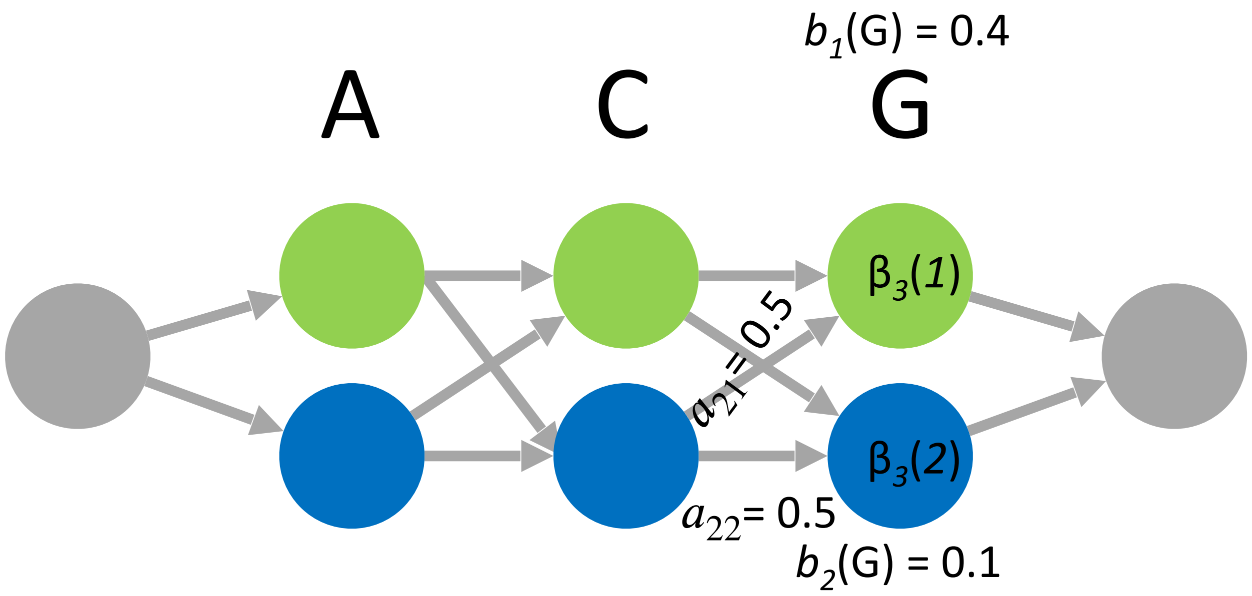
1. Initialization:

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N$$

2. Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \qquad 1 \leq t \leq T-1, 1 \leq j \leq N.$$

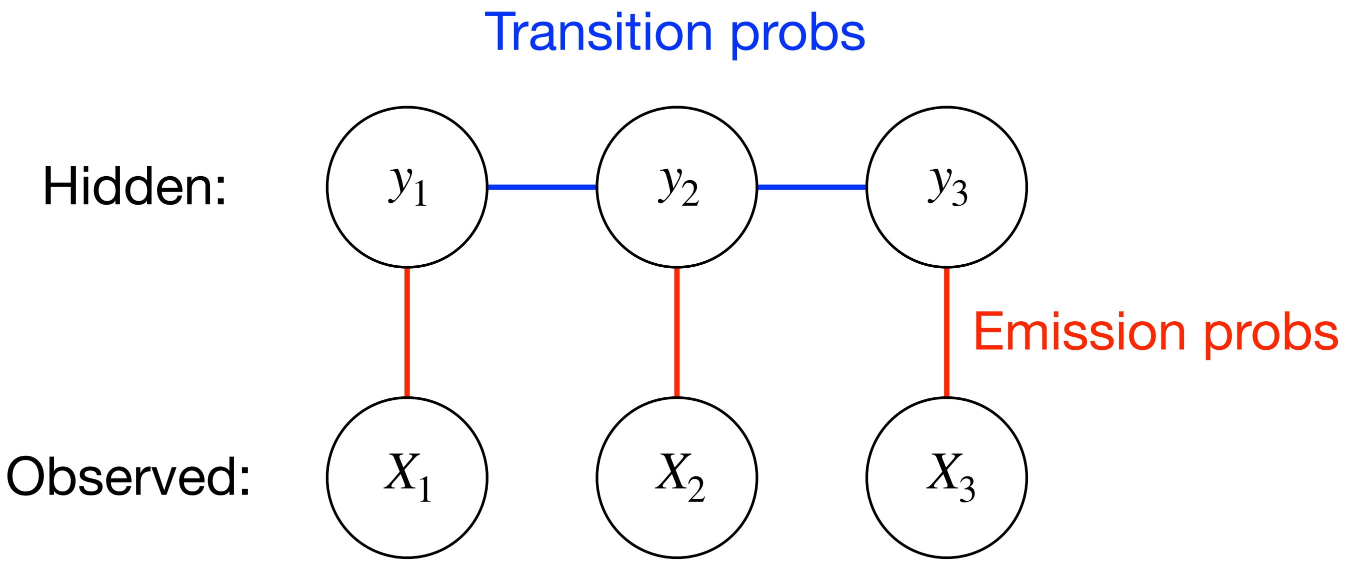
Build a dynamic programming table for these calculations

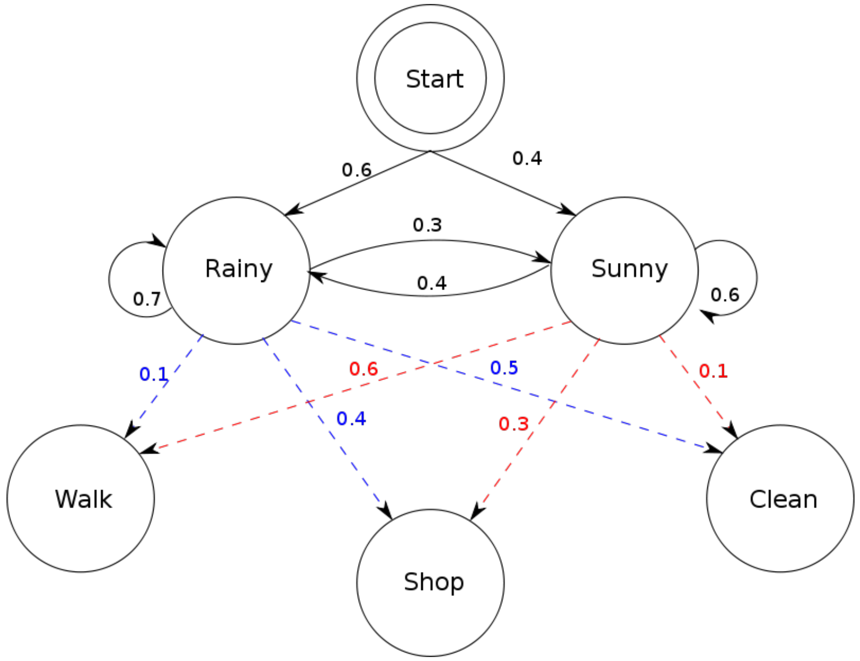


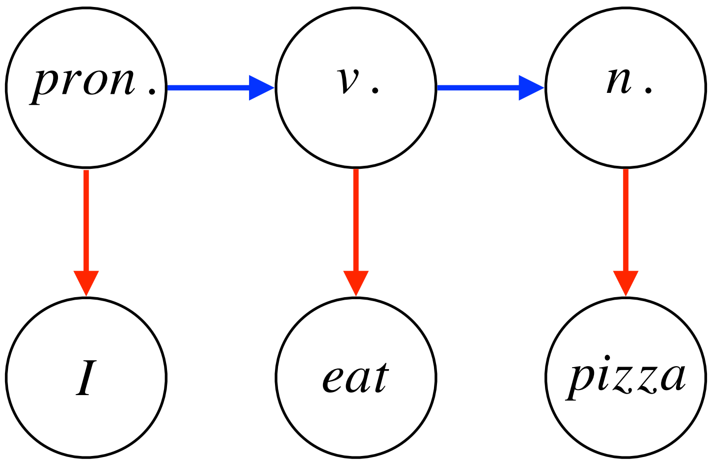
	$\beta_1(1)$	$\beta_2(1)$	$\beta_3(1)$
	A	C	G
State 1		0.28	1
State 2		0.25	1
	$\beta_1(2)$	$\beta_2(2)$	$\beta_3(2)$

HW8 questions?

HMM





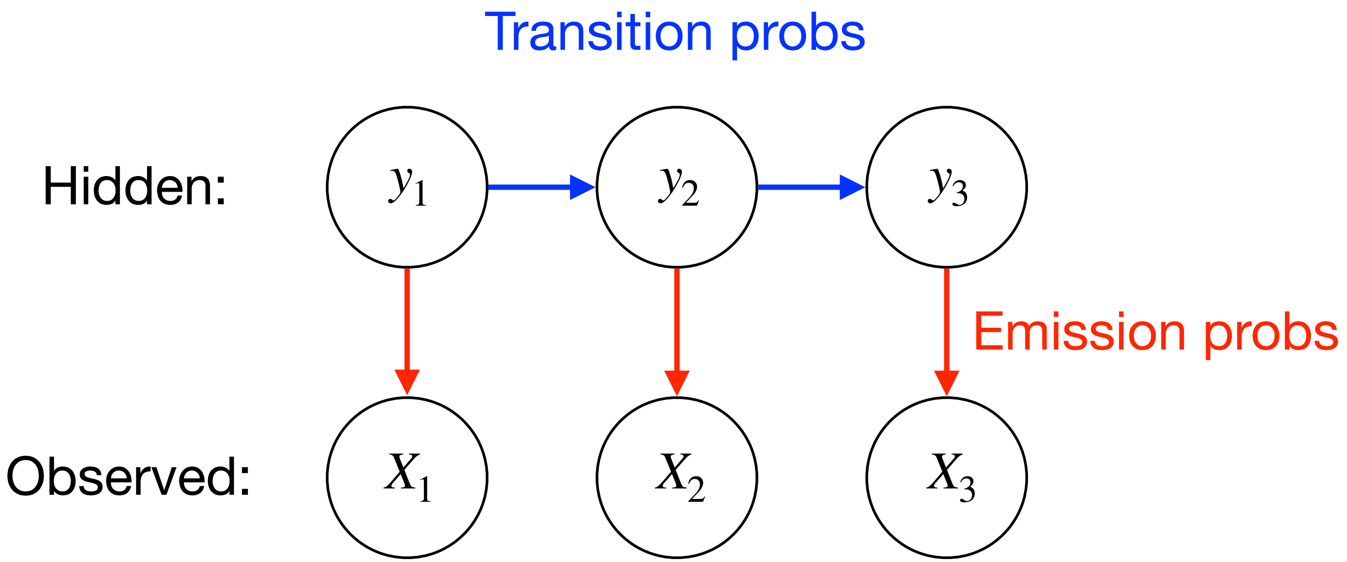


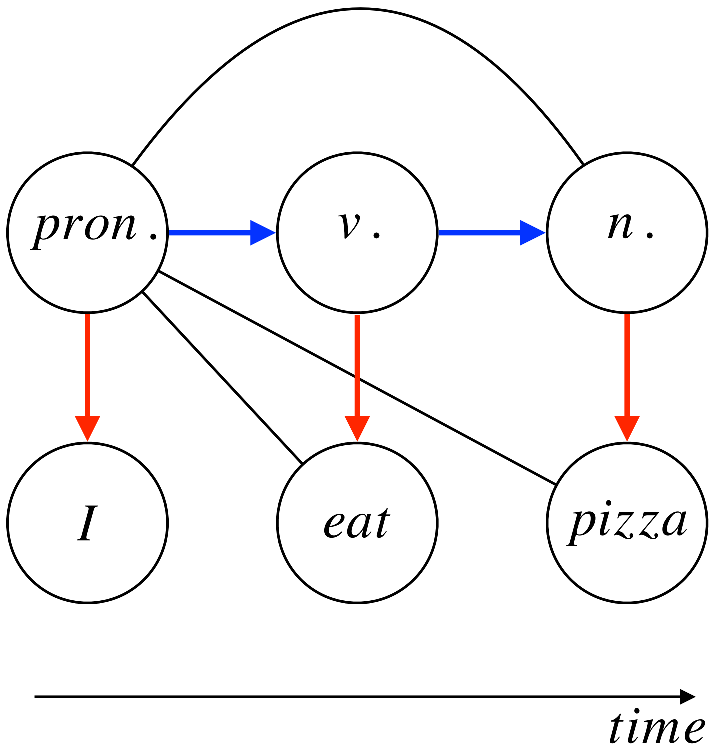
Joint probability

$$P(y, X) = \prod_{i=1}^{N} P(y_i \mid y_{i-1}) P(X_i \mid y_i)$$

Transition probs Emission probs

Limitations





Joint probability

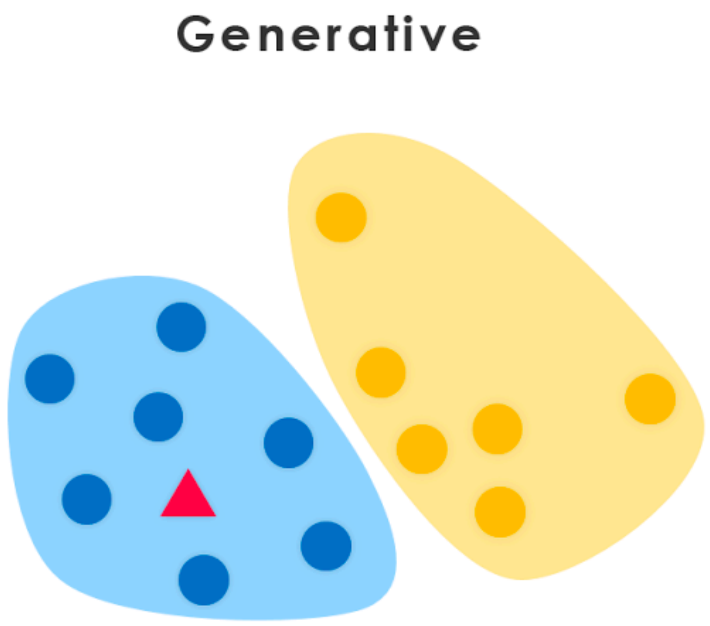
$$P(y, X) = \prod_{i=1}^{N} P(y_i \mid y_{i-1}) P(X_i \mid y_i)$$

Transition probs Emission probs

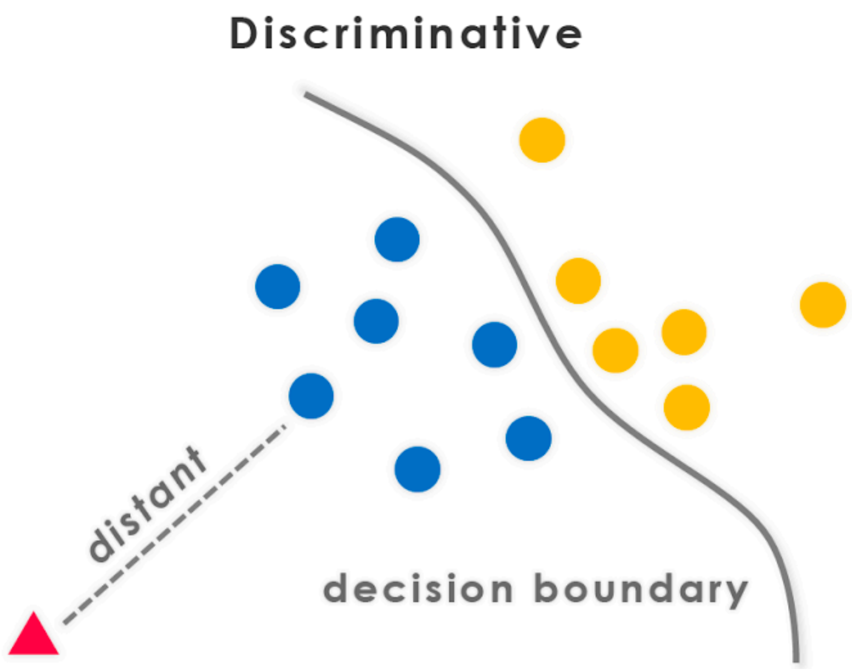
- Static transition/emission probs
- Limited dependences

Conditional random fields

- Hidden markov model
- **Conditional random fields**

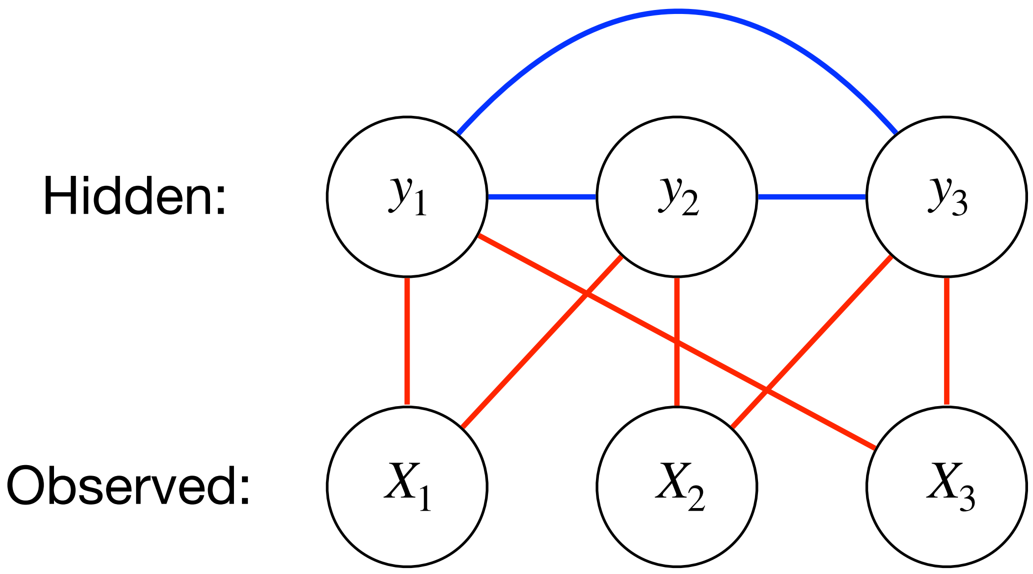


$P(y, X)$

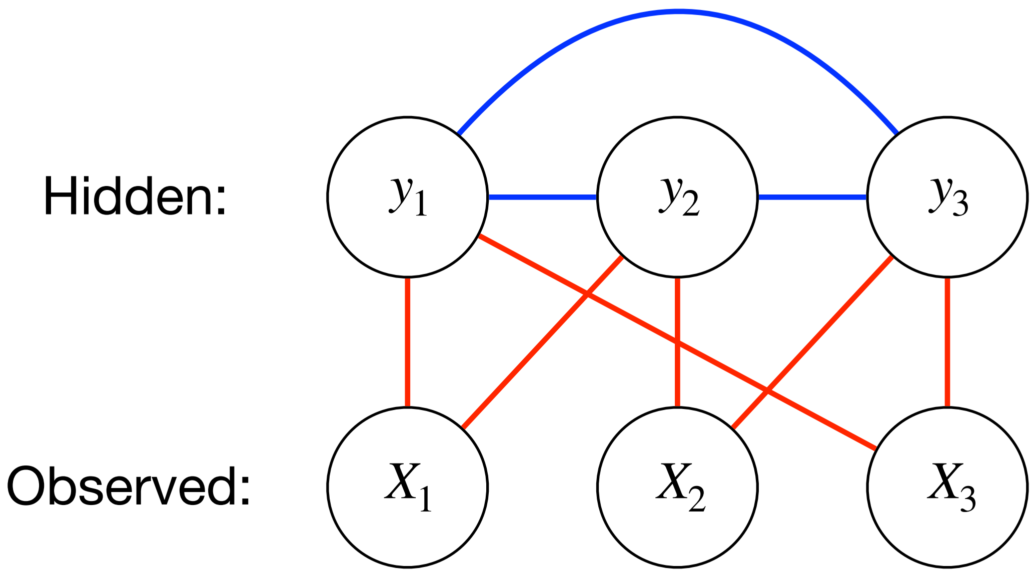


$P(y \mid X)$

Conditional random fields



A more general frameworks to capture the dependences between hidden states and observed states.



Linear chain CRF

Conditional random fields

2 The CRF Model

Let $x_{1:N}$ be the observations (e.g., words in a document), and $z_{1:N}$ the hidden labels (e.g., tags). A linear chain Conditional Random Field defines a *conditional probability* (whereas HMM defines the joint)

$$p(z_{1:N}|x_{1:N}) = \frac{1}{Z} \exp\left(\sum_{n=1}^{N} \sum_{i=1}^{F} \lambda_i f_i(z_{n-1}, z_n, x_{1:N}, n) \right). \quad (1)$$

Within the exp() function, we sum over $n = 1, \ldots, N$ word positions in the sequence. For each position, we sum over $i = 1, \ldots, F$ *weighted features*. The scalar λ_i is the weight for feature $f_i()$. The λ_i's are the *parameters* of the CRF model, and must be learned, similar to $\theta = \{\pi, \phi, A\}$ in HMMs.

3 Feature Functions

The feature functions are the key components of CRF. In our special case of linear-chain CRF, the general form of a feature function is $f_i(z_{n-1}, z_n, x_{1:N}, n)$, which looks at a pair of adjacent states z_{n-1}, z_n, the *whole* input sequence $x_{1:N}$, and where we are in the sequence. The feature functions produce a real value.

For example, we can define a simple feature function which produces binary values: it is 1 if the current word is John, and if the current state z_n is PERSON:

$$f_1(z_{n-1}, z_n, x_{1:N}, n) = \begin{cases} 1 & \text{if } z_n = \text{PERSON and } x_n = \text{John} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

https://pages.cs.wisc.edu/~jerryzhu/cs769/CRF.pdf