10-301/601: Introduction to Machine Learning Lecture 23: Hidden Markov Models

Henry Chai

7/19/23

Front Matter

- Announcements
  - PA5 released 7/13, due 7/20 (tomorrow) at 11:59 PM
  - PA6 released 7/20 (tomorrow), due 7/27 at 11:59 PM
- Recommended Readings
  - Murphy, Chapters 17.1 - 17.5

Recall: Hidden Markov Models

- Two types of variables: observations (observed) and states (hidden or latent)
  - Set of states usually pre-specified via domain expertise/prior knowledge: $\{s_1, \ldots, s_M\}$
  - Emission model:
    - Current observation is conditionally independent of all other variables given the current state: $P(X_t | Y_t)$
  - Transition model (Markov assumption):
    - Current state is conditionally independent of all earlier states given the previous state: $P(Y_t | Y_{t-1}, \ldots, Y_0) = P(Y_t | Y_{t-1})$

Hidden Markov Models: Outline

- How can we learn the conditional probabilities used by a hidden Markov model?
- What inference questions can we answer with a hidden Markov model?
  1. Computing the distribution of a single state (or a sequence of states) given a sequence of observations
  2. Finding the most-probable sequence of states given a sequence of observations
  3. Computing the probability of a sequence of observations

3 Inference Questions for HMMs

1. Marginal Computation: $P\left(Y_t = s_j \,\middle|\, \boldsymbol{x}^{(n)}\right)$ (or $P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$)

$$P\left(Y \middle| \boldsymbol{x}^{(n)}\right) = \frac{P\left(\boldsymbol{x}^{(n)} | Y\right) P(Y)}{P\left(\boldsymbol{x}^{(n)}\right)} = \frac{\prod_{t=1}^{T} P\left(\boldsymbol{x}_t^{(n)} \middle| Y_t\right) P(Y_t | Y_{t-1})}{P\left(\boldsymbol{x}^{(n)}\right)}$$

2. Decoding: $\hat{Y} = \underset{Y}{\operatorname{argmax}}\ P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$

3. Evaluation: $P\left(\boldsymbol{x}^{(n)}\right)$

$$P\left(\boldsymbol{x}^{(n)}\right) = \sum_{\mathcal{Y} \in \{\text{all possible sequences}\}} P\left(\boldsymbol{x}^{(n)} \middle| \mathcal{Y}\right) P(\mathcal{Y})$$

The Brute Force Algorithm

- Inputs: query $P(\boldsymbol{x}^{(n)})$, emission matrix A, transition matrix B
- Initialize $p = 0$
- For $\mathcal{Y} \in \{\text{all possible sequences}\}$
    - Compute the joint probability

$$P(\boldsymbol{x}^{(n)}, \mathcal{Y}) = P(\boldsymbol{x}^{(n)} | \mathcal{Y}) P(\mathcal{Y}) = \prod_{t=1}^{T} P\left(\boldsymbol{x}_t^{(n)} \middle| \mathcal{Y}_t\right) P(\mathcal{Y}_t | \mathcal{Y}_{t-1})$$

    - $p \mathrel{+}= P(\boldsymbol{x}^{(n)}, \mathcal{Y})$
- Return $p = P(\boldsymbol{x}^{(n)})$

When survey is active, respond at **pollev.com/301601polls**

Lecture 23 Polls

0 done

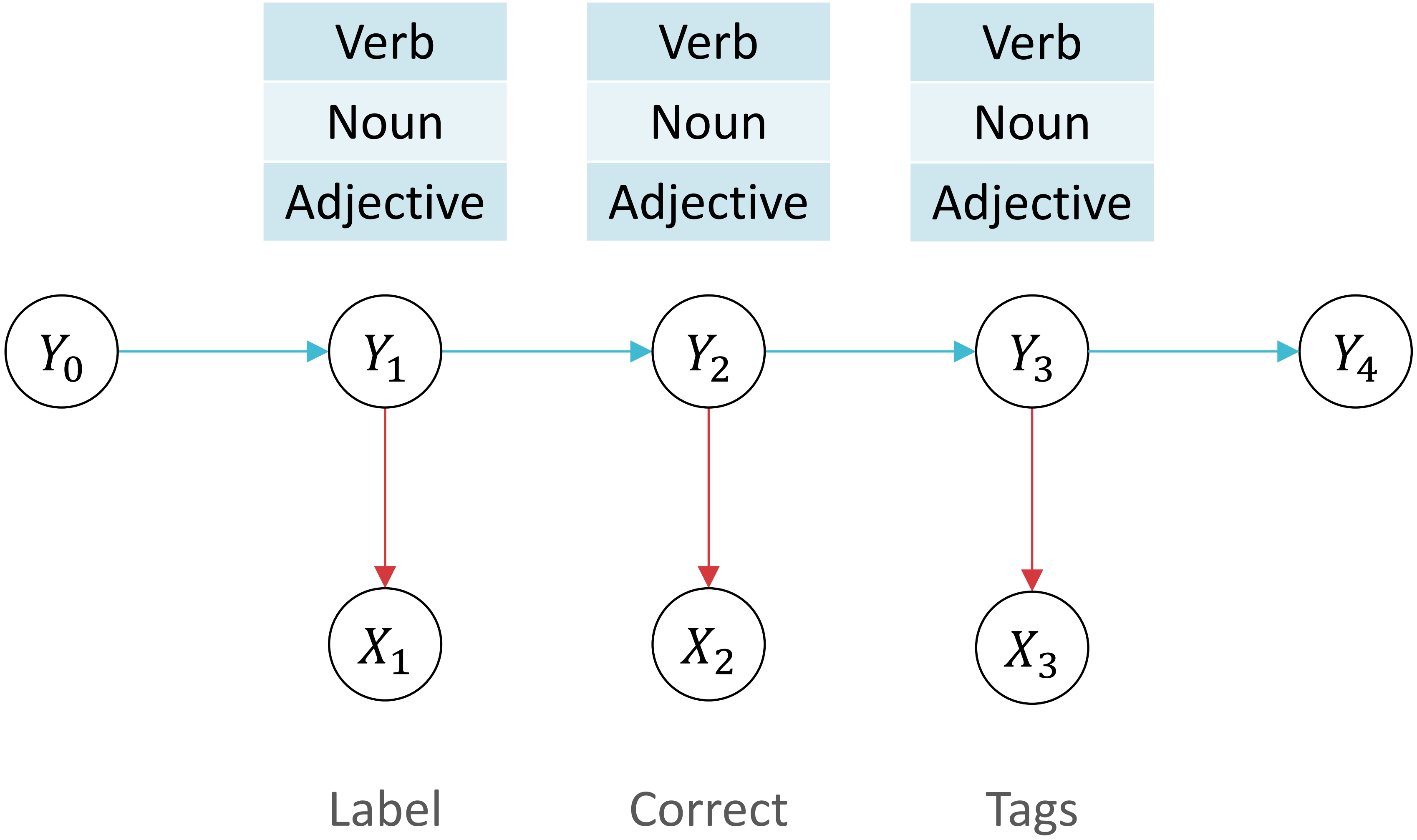
0 underway

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

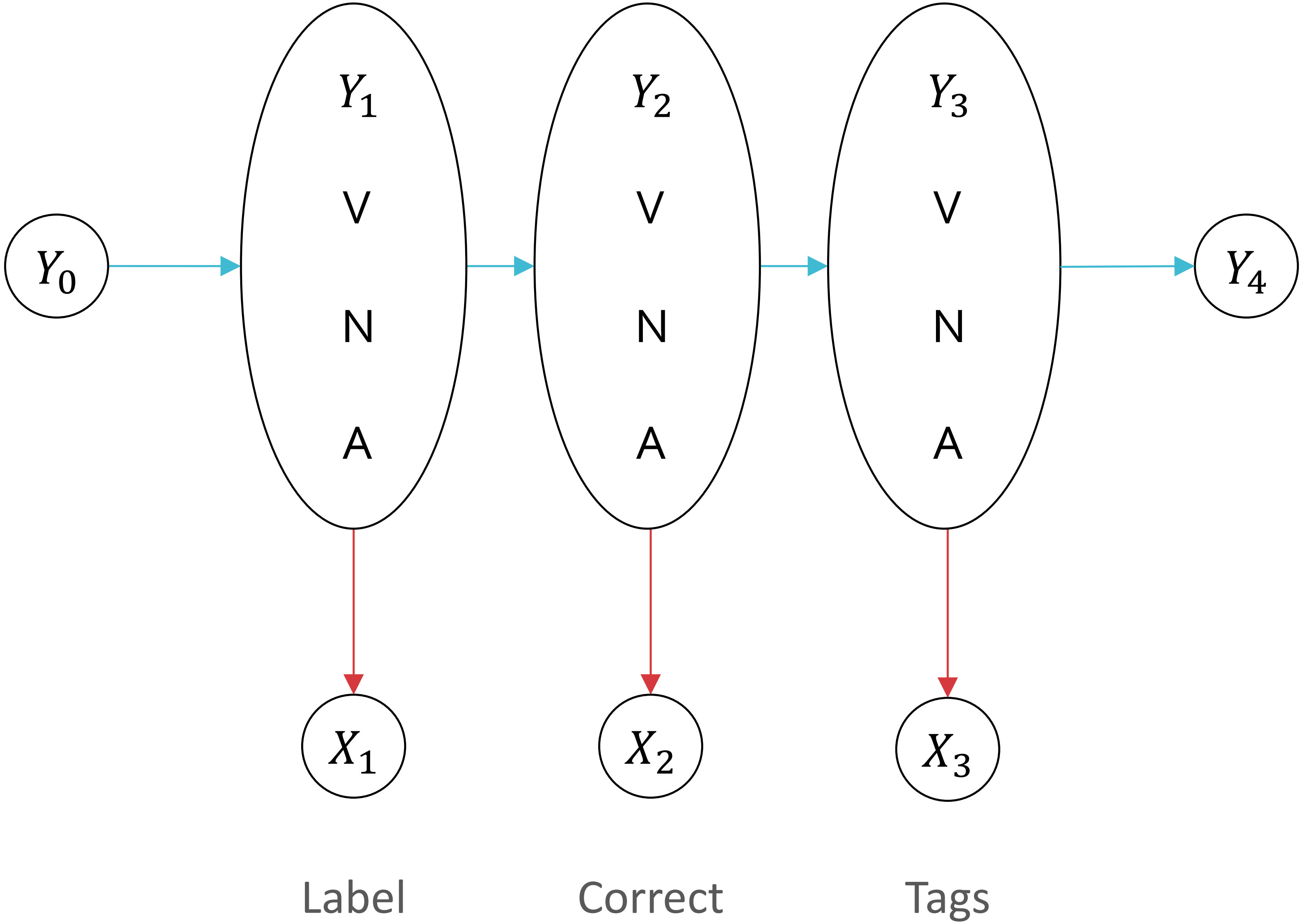
Given C possible observations and M possible states plus special START/END states, how many possible sequences of length T (not counting the START and END states) are there?

- TC
- TM
- T^M
- M^T

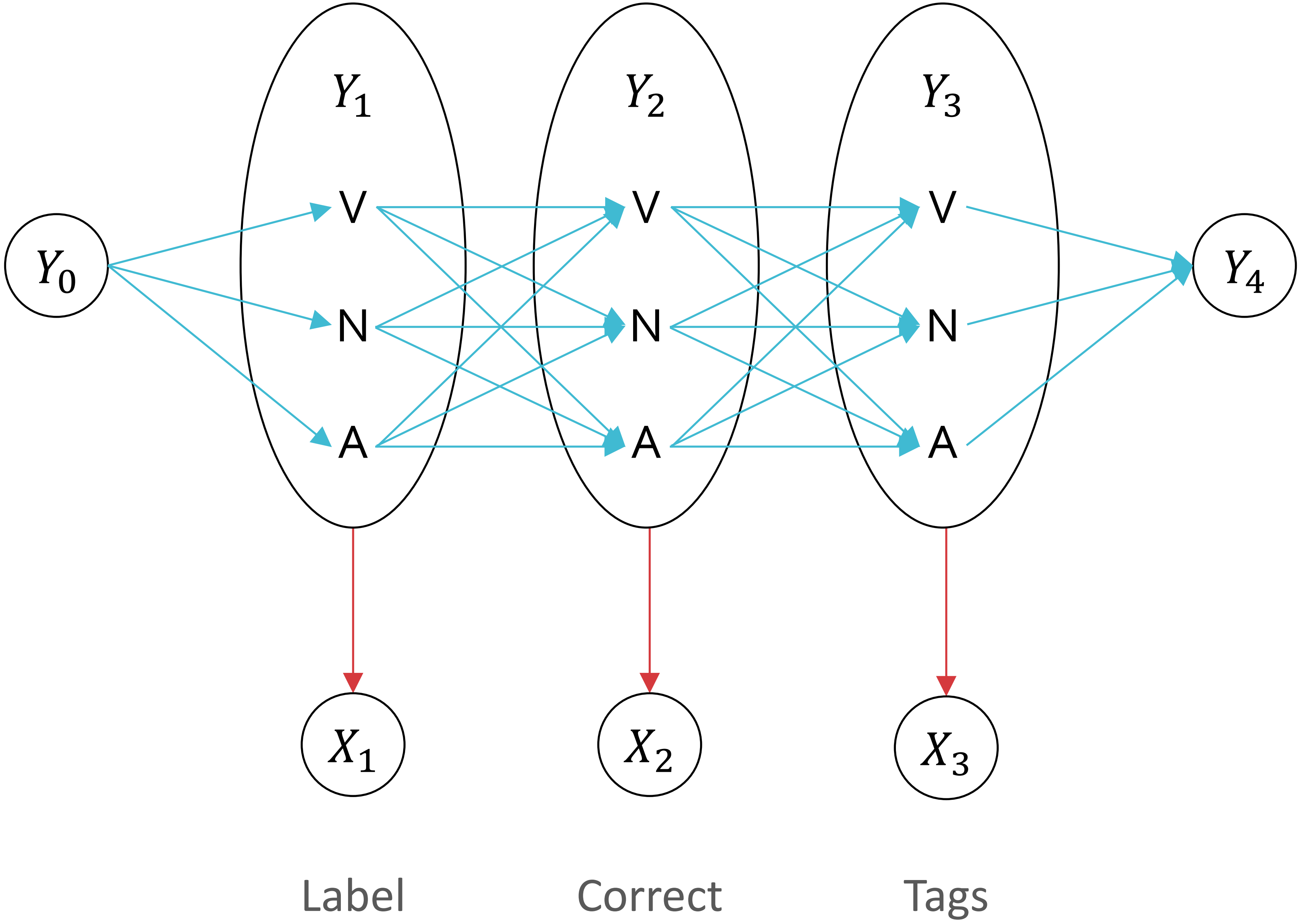
Inference with HMMs: PoS Tagging Example



Inference with HMMs: PoS Tagging Example



Inference with HMMs: PoS Tagging Example



3 Inference Questions for HMMs

1. Marginal Computation: $P\left(Y_t = s_j \,\middle|\, \boldsymbol{x}^{(n)}\right)$ (or $P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$)

$$P\left(Y_t = s_j \,\middle|\, \boldsymbol{x}^{(n)}\right) = \frac{P\left(Y_t = s_j, \boldsymbol{x}^{(n)}\right)}{P\left(\boldsymbol{x}^{(n)}\right)}$$

2. Decoding: $\hat{Y} = \underset{Y}{\operatorname{argmax}}\ P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$

3. Evaluation: $P\left(\boldsymbol{x}^{(n)}\right)$

$$P\left(\boldsymbol{x}^{(n)}\right) = \sum_{m=1}^{M} P\left(Y_t = s_m, \boldsymbol{x}^{(n)}\right)$$

Recursive Marginals

$$P\left(Y_t = s_j, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_T^{(n)}\right)$$

$$= P\left(\boldsymbol{x}_{t+1}^{(n)}, \dots, \boldsymbol{x}_T^{(n)} \middle| Y_t = s_j, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right) P\left(Y_t = s_j, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right)$$

By conditional independence assumptions

$$= P\left(\boldsymbol{x}_{t+1}^{(n)}, \dots, \boldsymbol{x}_T^{(n)} \middle| Y_t = s_j\right) P\left(Y_t = s_j, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right)$$

$$\coloneqq \beta_t(j)\alpha_t(j)$$

Can be computed recursively (forward algorithm)

Can be computed recursively (backward algorithm)

Forward Algorithm

$$\alpha_t(j) \coloneqq P\left(Y_t = s_j, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right)$$

$$= \sum_{m=1}^{M} P\left(Y_t = s_j, Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_t^{(n)} | Y_t = s_j, Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_{t-1}^{(n)}\right) *$$

$$P\left(Y_t = s_j, Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_{t-1}^{(n)}\right)$$

Forward Algorithm

$$\alpha_t(j) \coloneqq P\left(Y_t = s_j, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right)$$

$$= \sum_{m=1}^{M} P\left(Y_t = s_j, Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_t^{(n)} | Y_t = s_j, Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_{t-1}^{(n)}\right) *$$

$$P\left(Y_t = s_j \middle| Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_{t-1}^{(n)}\right) P\left(Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_{t-1}^{(n)}\right)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_t^{(n)} | Y_t = s_j\right) P\left(Y_t = s_j \middle| Y_{t-1} = s_m\right) P\left(Y_{t-1} = s_m, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_{t-1}^{(n)}\right)$$

$$= P\left(\boldsymbol{x}_t^{(n)} | Y_t = s_j\right) \sum_{m=1}^{M} P\left(Y_t = s_j \middle| Y_{t-1} = s_m\right) \alpha_{t-1}(m)$$

- Base case: $\alpha_0(\text{START}) = 1$

Backward Algorithm

$$\beta_t(j) \coloneqq P\left(\boldsymbol{x}_{t+1}^{(n)}, \dots, \boldsymbol{x}_T^{(n)} \middle| Y_t = s_j\right)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_{t+1}^{(n)}, \dots, \boldsymbol{x}_T^{(n)}, Y_{t+1} = s_m \middle| Y_t = s_j\right)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_{t+2}^{(n)}, \dots, \boldsymbol{x}_T^{(n)} | Y_t = s_j, \boldsymbol{x}_{t+1}^{(n)}, Y_{t+1} = s_m\right) *$$

$$P\left(\boldsymbol{x}_{t+1}^{(n)}, Y_{t+1} = s_m \middle| Y_t = s_j\right)$$

Backward Algorithm

$$\beta_t(j) \coloneqq P\left(\boldsymbol{x}_{t+1}^{(n)}, \dots, \boldsymbol{x}_T^{(n)} \middle| Y_t = s_j\right)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_{t+1}^{(n)}, \dots, \boldsymbol{x}_T^{(n)}, Y_{t+1} = s_m \middle| Y_t = s_j\right)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_{t+2}^{(n)}, \dots, \boldsymbol{x}_T^{(n)} | Y_t = s_j, \boldsymbol{x}_{t+1}^{(n)}, Y_{t+1} = s_m\right) *$$
$$P\left(\boldsymbol{x}_{t+1}^{(n)} \middle| Y_{t+1} = s_m, Y_t = s_j\right) P(Y_{t+1} = s_m | Y_t = s_j)$$

$$= \sum_{m=1}^{M} P\left(\boldsymbol{x}_{t+2}^{(n)}, \dots, \boldsymbol{x}_T^{(n)} | Y_{t+1} = s_m\right)$$
$$P\left(\boldsymbol{x}_{t+1}^{(n)} \middle| Y_{t+1} = s_m\right) P(Y_{t+1} = s_m | Y_t = s_j)$$

$$= \sum_{m=1}^{M} \beta_{t+1}(m) P\left(\boldsymbol{x}_{t+1}^{(n)} \middle| Y_{t+1} = s_m\right) P(Y_{t+1} = s_m | Y_t = s_j)$$

- Base case: $\beta_{T+1}(\text{END}) = 1$

The Forward-Backward Algorithm

- Inputs: query $P\left(Y_t = s_j \,\middle|\, \boldsymbol{x}^{(n)}\right)$, emission matrix A, transition matrix B
- Initialize $\alpha_0(\text{START}) = 1$ and $\beta_{T+1}(\text{END}) = 1$
- For $\tau = 1, \dots, T$
  - For $m = 1, \dots, M$

$$\alpha_\tau(m) = P\left(\boldsymbol{x}_\tau^{(n)} | Y_\tau = s_m\right) \sum_{k=1}^{M} P(Y_\tau = s_m | Y_{\tau-1} = s_k) \alpha_{\tau-1}(k)$$

- For $\tau = T, \dots, 1$
  - For $m = 1, \dots, M$

$$\beta_\tau(m) = \sum_{k=1}^{M} \beta_{\tau+1}(k) P\left(\boldsymbol{x}_{\tau+1}^{(n)} \middle| Y_{\tau+1} = s_k\right) P(Y_{\tau+1} = s_k | Y_\tau = s_m)$$

- Return $P\left(Y_t = s_j \,\middle|\, \boldsymbol{x}^{(n)}\right) = \frac{P\left(Y_t = s_j, \boldsymbol{x}^{(n)}\right)}{P\left(\boldsymbol{x}^{(n)}\right)} = \frac{\beta_t(j)\alpha_t(j)}{\sum_{m=1}^{M} \beta_t(m)\alpha_t(m)}$

Given C possible observations and M possible states plus special START/END states, what is the runtime of the forward-backward algorithm on sequences of length T?

- $O(TM)$
- $O(T^2M)$
- $O(TM^2)$
- $O(T^2M^2)$

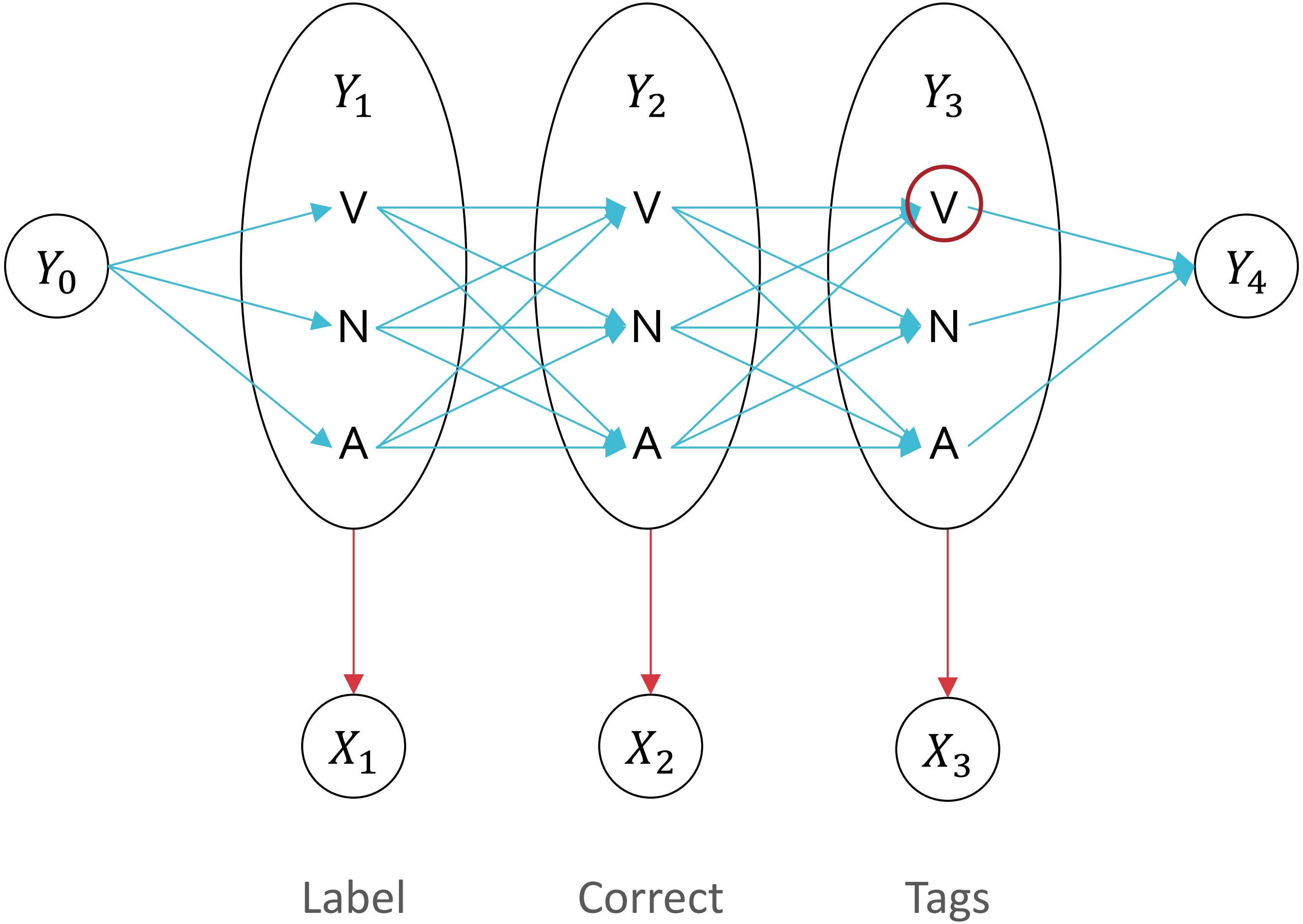
Most Probable State Sequence

Decoding: $\hat{Y} = \underset{Y}{\operatorname{argmax}} \; P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$

$$\omega_t(j) \coloneqq \max_{\mathcal{Y} \in \{\text{all possible sequences of } t-1 \text{ states}\}} P\left(\mathcal{Y}, Y_t = s_j, \boldsymbol{x}_1^{(n)}, \dots, \boldsymbol{x}_t^{(n)}\right)$$

$=$ the probability of the most probable sequence of t states that ends in s_j, conditioned on the first t observations

$\omega_3(V)$



$\omega_3(V)$

$= \max\{$

$\omega_2(V) P(Y_3 = V | Y_2 = V)$

$P\left(x_3^{(n)} \middle| Y_3 = V\right),$

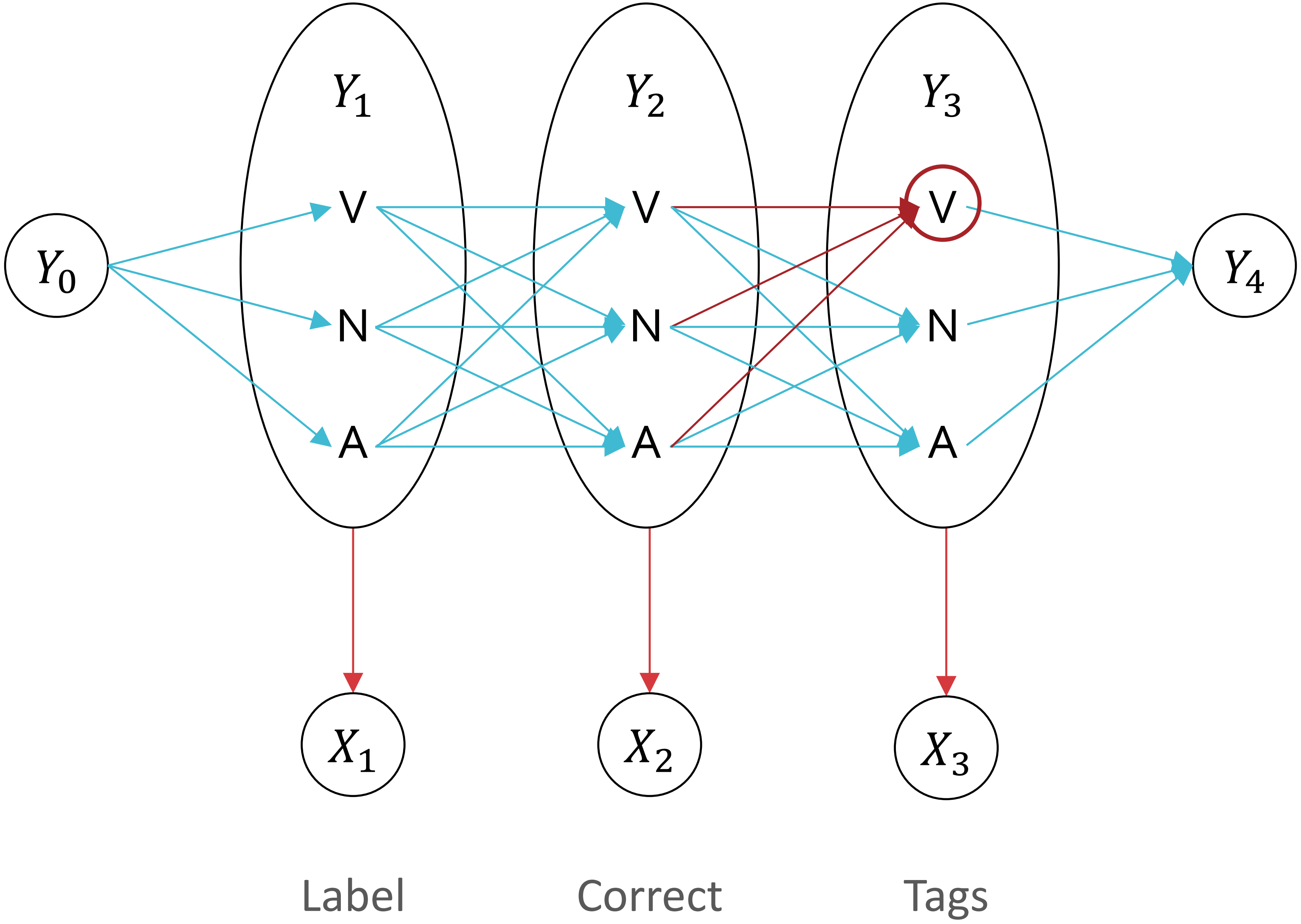
$\omega_2(N) P(Y_3 = V | Y_2 = N)$

$P\left(x_3^{(n)} \middle| Y_3 = V\right),$

$\omega_2(A) P(Y_3 = V | Y_2 = A)$

$P\left(x_3^{(n)} \middle| Y_3 = V\right),$

$\}$



$\omega_3(V)$

$= \max\{$

$\omega_2(V)P(Y_3 = V|Y_2 = V)$

$P\left(x_3^{(n)} \middle| Y_3 = V\right),$

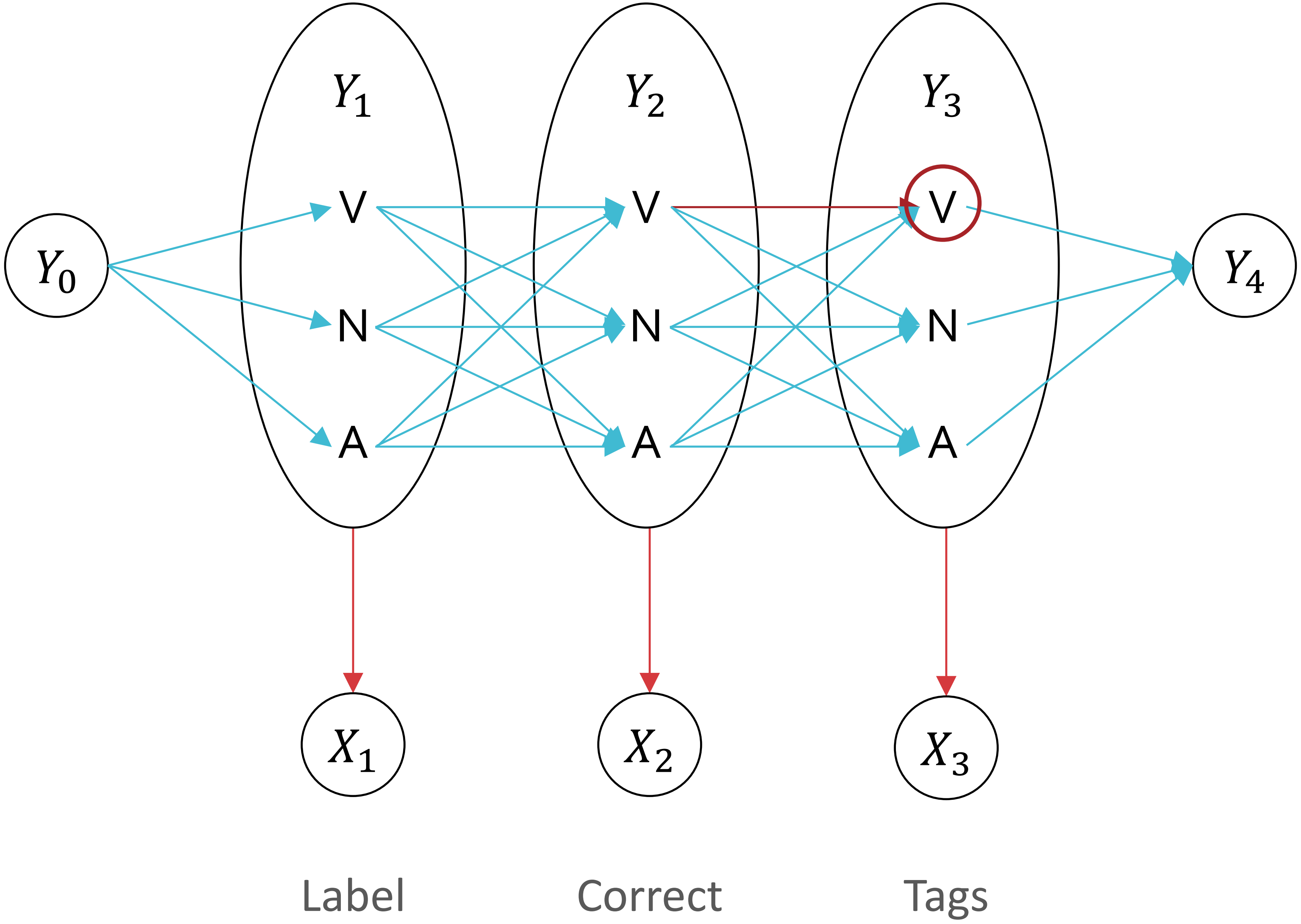
$\omega_2(N)P(Y_3 = V|Y_2 = N)$

$P\left(x_3^{(n)} \middle| Y_3 = V\right),$

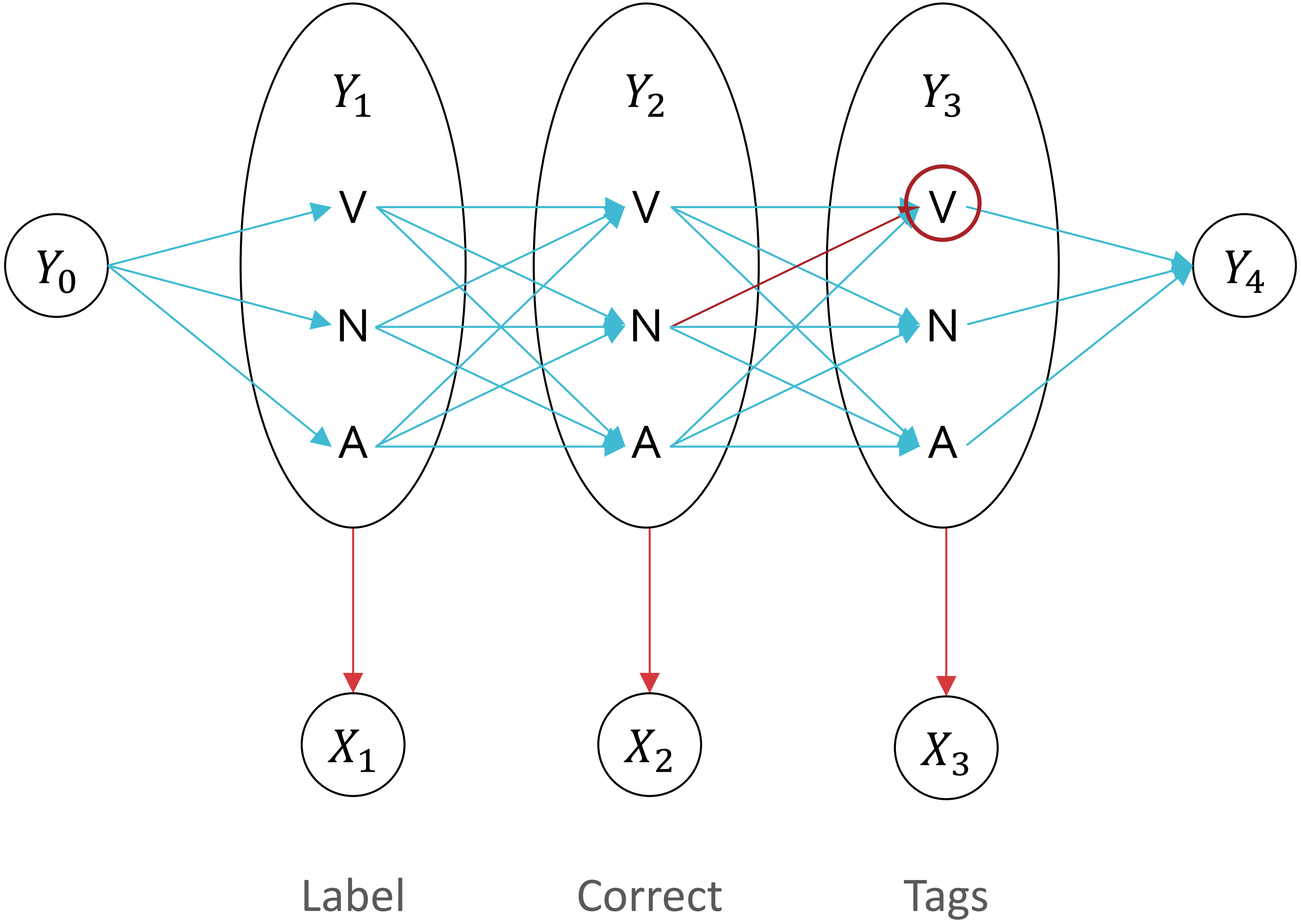
$\omega_2(A)P(Y_3 = V|Y_2 = A)$

$P\left(x_3^{(n)} \middle| Y_3 = V\right),$

$\}$



$\omega_3(V)$
$= \max\{$
$\omega_2(V) P(Y_3 = V | Y_2 = V)$
$P\left(x_3^{(n)} \middle| Y_3 = V\right),$
$\omega_2(N) P(Y_3 = V | Y_2 = N)$
$P\left(x_3^{(n)} \middle| Y_3 = V\right),$
$\omega_2(A) P(Y_3 = V | Y_2 = A)$
$P\left(x_3^{(n)} \middle| Y_3 = V\right),$
$\}$



$\omega_3(V)$

$= \max\{$

$\omega_2(V)P(Y_3 = V|Y_2 = V)$

$P\left(x_3^{(n)}\middle|Y_3 = V\right),$

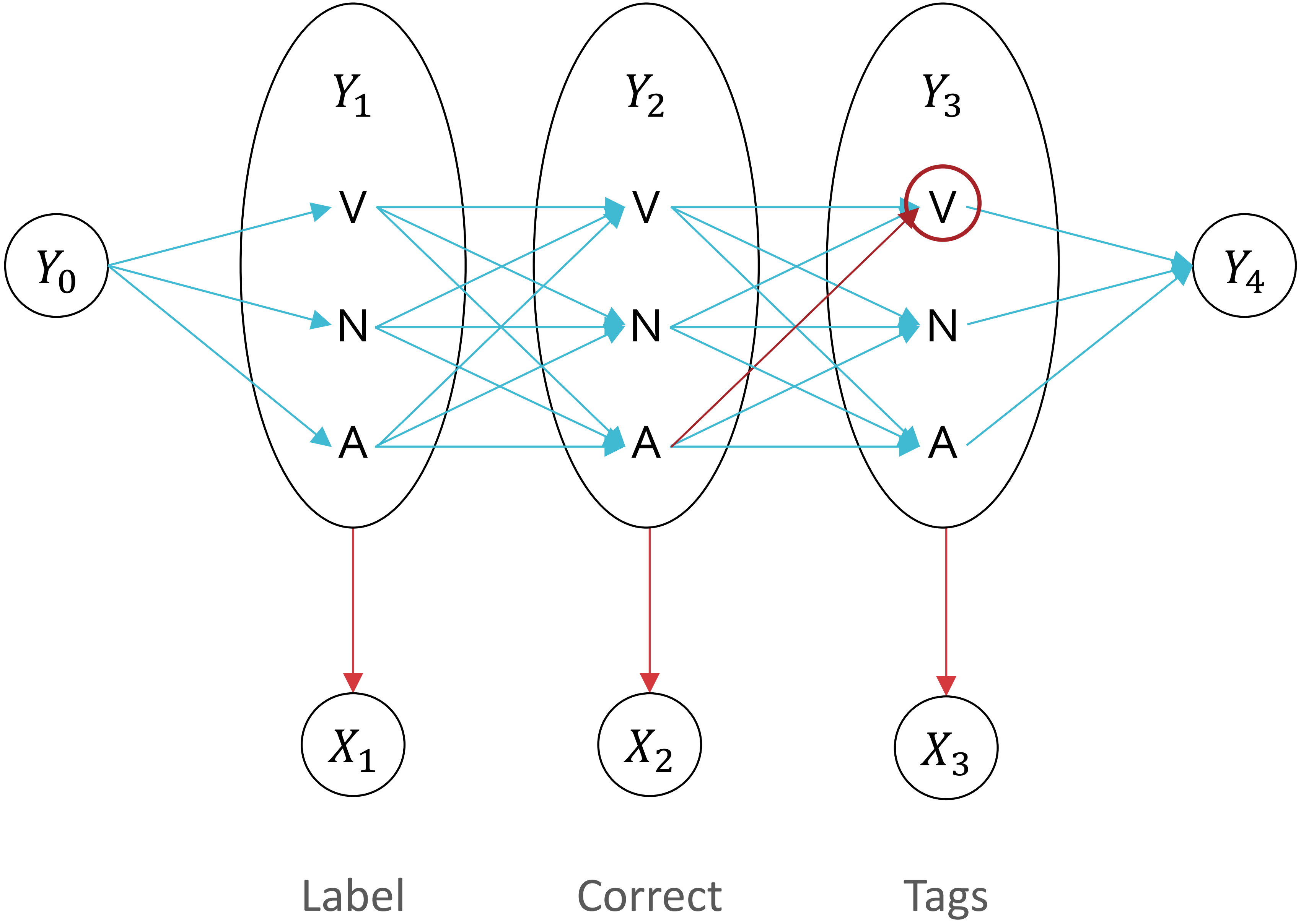
$\omega_2(N)P(Y_3 = V|Y_2 = N)$

$P\left(x_3^{(n)}\middle|Y_3 = V\right),$

$\omega_2(A)P(Y_3 = V|Y_2 = A)$

$P\left(x_3^{(n)}\middle|Y_3 = V\right),$

$\}$



Most Probable State Sequence

Decoding: $\hat{Y} = \underset{Y}{\operatorname{argmax}}\ P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$

$$\omega_t(j) \coloneqq \max_{\mathcal{Y} \in \{\text{all possible sequences of } t-1 \text{ states}\}} P\left(\mathcal{Y}, Y_t = s_j, \boldsymbol{x}_1^{(n)}, \ldots, \boldsymbol{x}_t^{(n)}\right)$$

$=$ the probability of the most probable sequence of t states that ends in s_j, conditioned on the first t observations

$$= \max_{m \in \{1,\ldots,M\}} \omega_{t-1}(m)\, P\left(Y_t = s_j | Y_{t-1} = s_m\right) P\left(\boldsymbol{x}_t^{(n)} | Y_t = s_j\right)$$

The Viterbi Algorithm

- Inputs: observations $\boldsymbol{x}^{(n)}$, emission matrix A, transition matrix B
- Initialize $\omega_0(\text{START}) = 1$
- For $\tau = 1, \ldots, T+1$
  - For $m = 1, \ldots, M$

$$\omega_\tau(m) = \max_{k \in \{1,\ldots,M\}} P\left(\boldsymbol{x}_\tau^{(n)} | Y_\tau = s_m\right) P(Y_\tau = s_m | Y_{\tau-1} = s_k) \omega_{\tau-1}(k)$$

$$\rho_\tau(m) = \underset{k \in \{1,\ldots,M\}}{\text{argmax}}\ P\left(\boldsymbol{x}_\tau^{(n)} | Y_\tau = s_m\right) P(Y_\tau = s_m | Y_{\tau-1} = s_k) \omega_{\tau-1}(k)$$

- Return the most probable assignment given $\boldsymbol{x}^{(n)}$:
  - $\hat{Y}_T = \rho_{T+1}(\text{END})$
  - For $\tau = T-1, \ldots, 1$
    - $\hat{Y}_\tau = \rho_{\tau+1}(\hat{Y}_{\tau+1})$

~~3~~ 4 Inference Questions for HMMs

1. Marginal Computation: $P\left(Y_t = s_j \middle| \boldsymbol{x}^{(n)}\right)$ (or $P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$)

$$P\left(Y \middle| \boldsymbol{x}^{(n)}\right) = \frac{P\left(\boldsymbol{x}^{(n)} | Y\right) P(Y)}{P\left(\boldsymbol{x}^{(n)}\right)} = \frac{\prod_{t=1}^{T} P\left(\boldsymbol{x}_t^{(n)} \middle| Y_t\right) P(Y_t | Y_{t-1})}{P\left(\boldsymbol{x}^{(n)}\right)}$$

2. Viterbi Decoding: $\hat{Y} = \underset{Y}{\operatorname{argmax}} \; P\left(Y \middle| \boldsymbol{x}^{(n)}\right)$

3. Evaluation: $P\left(\boldsymbol{x}^{(n)}\right)$

$$P\left(\boldsymbol{x}^{(n)}\right) = \sum_{\mathcal{Y} \in \{\text{all possible sequences}\}} P\left(\boldsymbol{x}^{(n)} \middle| \mathcal{Y}\right) P(\mathcal{Y})$$

4. Minimum Bayes Risk (MBR) Decoding:

$$\hat{Y} = \underset{Y}{\operatorname{argmin}} \; \mathbb{E}_{Y' \sim P_{A,B}\left(\cdot \middle| \boldsymbol{x}^{(n)}\right)}[\ell(Y, Y')]$$

Minimum Bayes Risk Decoding

- The learned parameters A and B induce a probability distribution or belief over sequences of states $P_{A,B}\left(Y \middle| \boldsymbol{x}^{(n)}\right)$
- Given a loss function, $\ell(Y, Y')$, find the sequence of states that minimizes our expected loss *under our current belief*

$$\hat{Y} = \underset{Y}{\operatorname{argmin}} \; \mathbb{E}_{Y' \sim P_{A,B}\left(\cdot \middle| \boldsymbol{x}^{(n)}\right)}[\ell(Y, Y')]$$

$$= \underset{Y}{\operatorname{argmin}} \sum_{Y'} P_{A,B}\left(Y' \middle| \boldsymbol{x}^{(n)}\right) \ell(Y, Y')$$

Minimum Bayes Risk Decoding: Example

- If $\ell(Y, Y')$ is the 0-1 loss

$$\ell(Y, Y') = 1 - \mathbb{1}(Y = Y')$$

$$\hat{Y} = \underset{Y}{\operatorname{argmin}} \sum_{Y'} P_{A,B}\left(Y' \middle| \boldsymbol{x}^{(n)}\right)\left(1 - \mathbb{1}(Y = Y')\right)$$

$$= \underset{Y}{\operatorname{argmin}} -\sum_{Y'} P_{A,B}\left(Y' \middle| \boldsymbol{x}^{(n)}\right)\mathbb{1}(Y = Y')$$

$$= \underset{Y}{\operatorname{argmax}}\ P_{A,B}\left(Y \middle| \boldsymbol{x}^{(n)}\right)$$

Minimum Bayes Risk Decoding: Example

- If $\ell(Y, Y')$ is the Hamming loss

$$\ell(Y, Y') = \sum_{t=1}^{T} 1 - \mathbb{1}(Y_t = Y_t')$$

$$\hat{Y}_t = \underset{Y_t}{\operatorname{argmax}} \; P_{A,B}\left(Y_t \middle| \boldsymbol{x}^{(n)}\right)$$

- Computes the most likely state at each time step using the marginals from the forward-backward algorithm

Key Takeaways

- Because of their well-behaved graphical structure, inference in HMMs is tractable via dynamic programming
  - Forward-backward algorithm for computing marginal distributions
  - Viterbi algorithm for computing most probable sequence of states