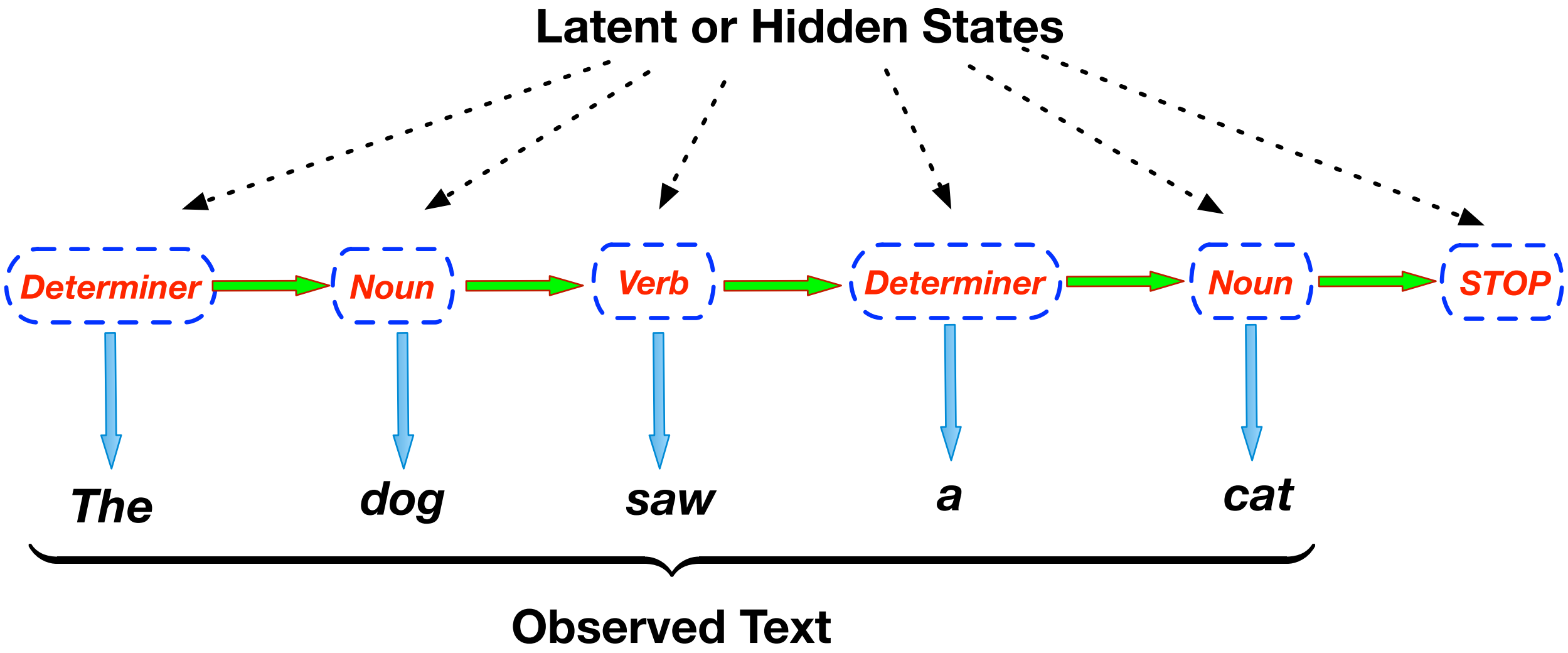
Special Topics in Natural Language Processing

CS698O

Ashutosh Modi
CSE Department, IIT Kanpur

Lecture 12: Sequence Prediction 3
Jan 31, 2020

Hidden Markov Models (HMM)



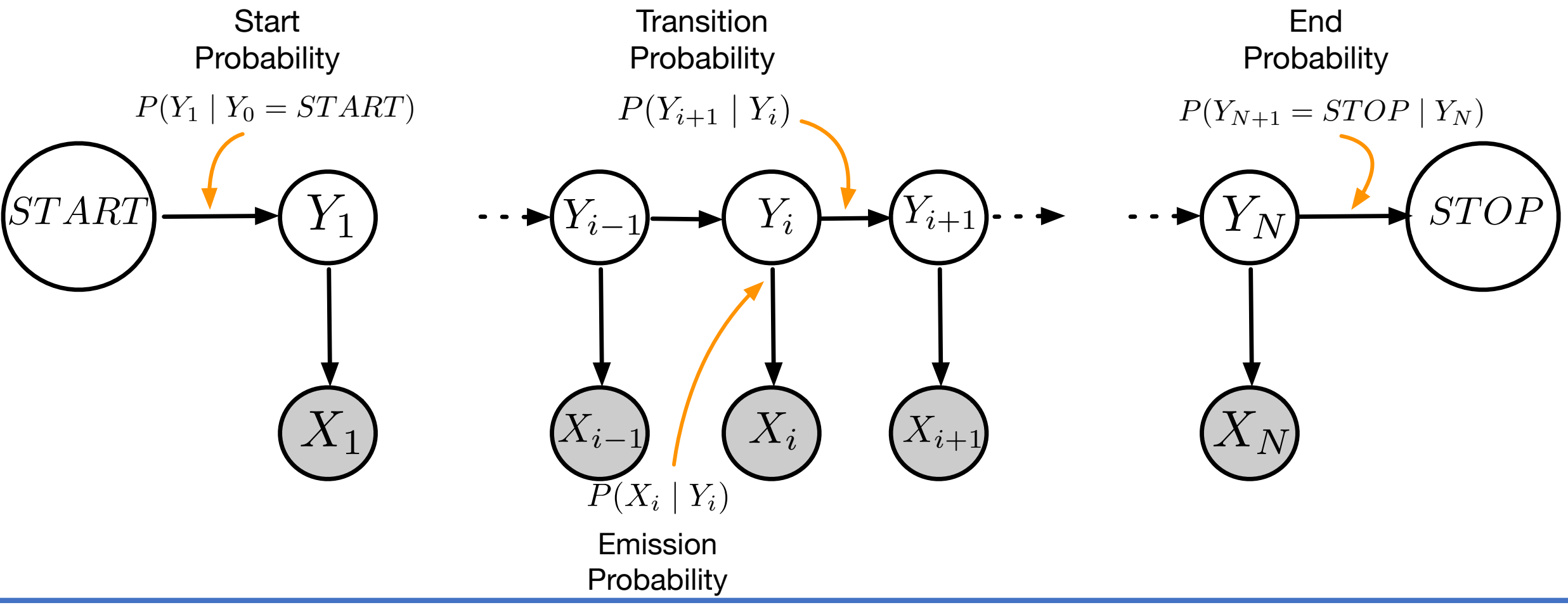
HMM Setting

Set of States (Λ)

$$\{c_0 = START, c_1, c_2, ..., c_K, c_{K+1} = STOP\}$$

Set of Observations

$$\{w_1, w_2, ...w_J\}$$



HMM Joint Distribution

- We have a generative model and are interested in the joint distribution

$$P(X, Y) = P\left(X_1 = x_1, \ldots, X_N = x_N, Y_1 = y_1, \ldots, Y_N = y_N\right)$$

$$P\left(X_1 = x_1, \ldots, X_N = x_N, Y_1 = y_1, \ldots, Y_N = y_N\right) =$$

$$P\left(y_1 | START\right) \times \left(\prod_{i=1}^{N-1} P\left(y_{i+1} | y_i\right)\right) \times \left(\prod_{i=1}^{N} P\left(x_i | y_i\right)\right) \times P\left(STOP | y_N\right)$$

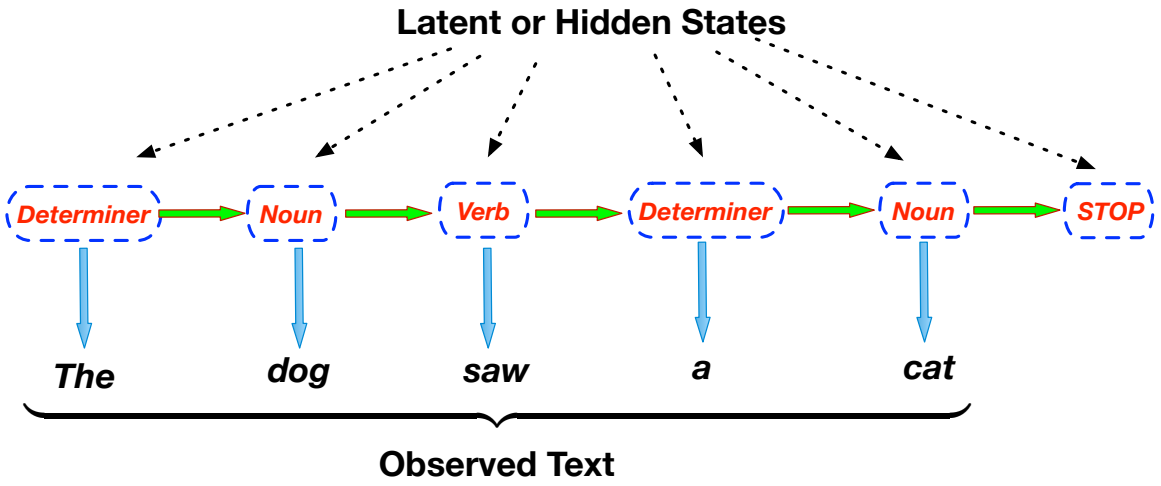
Initial / Start Probability

Transition Probability

Emission Probability

End / Final Probability

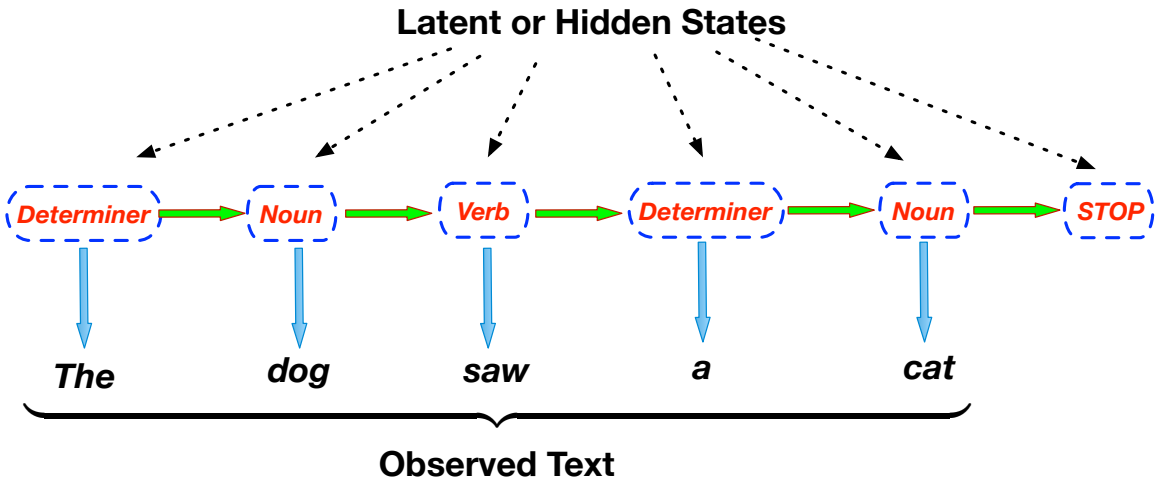
Hidden Markov Models (HMM)



Three Questions:

1. ***Learning Problem***: How do we estimate the parameters of distributions?
2. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?
3. ***Likelihood Problem***: Given the parameters of the distributions what is the probability of the observed sequence?

Hidden Markov Models (HMM)



Three Questions:

1. ***Learning Problem***: How do we estimate the parameters of distributions?
2. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?
3. ***Likelihood Problem***: Given the parameters of the distributions what is the probability of the observed sequence?

Learning Problem

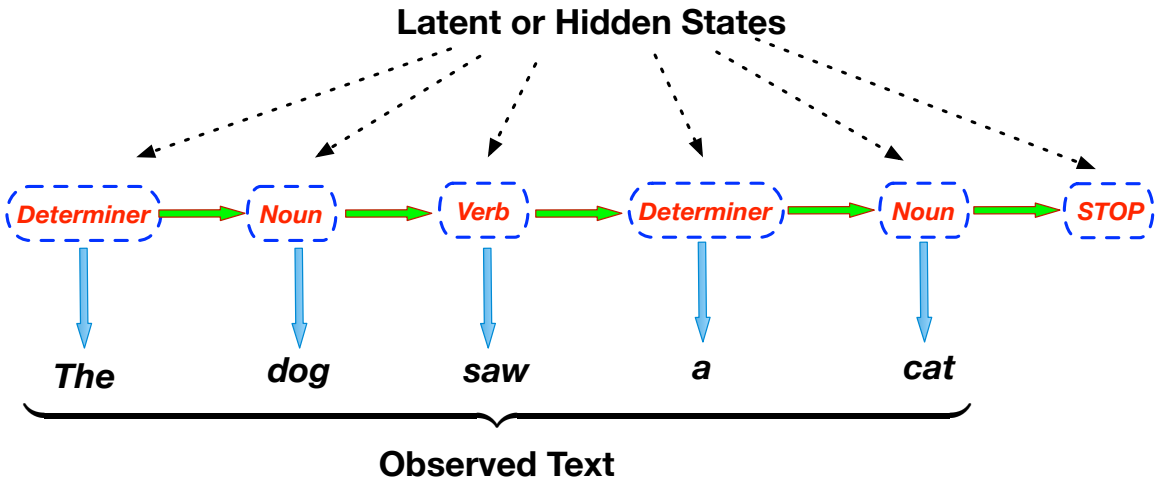
$$P_{\text{init}}\ (y_1 = c_k | START) = \frac{Count(y_1 = c_k)}{\left(\sum_{l=1}^{K} Count(y_1 = c_l)\right) = M}$$

$$P_{\text{trans}}\ (y_{i+1} = c_k | y_i = c_l) = \frac{Count(c_k, c_l)}{Count(c_l)}$$

$$P_{\text{emiss}}\ (x_i = w_j | y_i = c_k) = \frac{Count(w_j, c_k)}{Count(c_k)}$$

$$P_{\text{final}}\ (STOP | y_N = c_k) = \frac{Count(y_N = c_k)}{\left(\sum_{l=1}^{K} Count(y_N = c_l)\right) = M}$$

Hidden Markov Models (HMM)



Three Questions:

1. ***Learning Problem***: How do we estimate the parameters of distributions?
2. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?
3. ***Likelihood Problem***: Given the parameters of the distributions what is the probability of the observed sequence?

Decoding Problem

- Two ways possible

Decoding Problem

- Two ways possible
  - **Posterior Decoding**: Maximize probability of each state

$$y_i^* = \arg\max_{y_i \in \Lambda} P\left(Y_i = y_i | X_1 = x_1, \ldots, X_N = x_N\right)$$

Decoding Problem

- Two ways possible
  - **Posterior Decoding**: Maximize probability of each state

$$y_i^* = \arg\max_{y_i \in \Lambda} P\left(Y_i = y_i | X_1 = x_1, \ldots, X_N = x_N\right)$$

  - **Viterbi Decoding**: Maximize the probability of the entire sequence

$$y^* = \arg\max_{y = y_1 \ldots y_N} P\left(Y_1 = y_1, \ldots, Y_N = y_N | X_1 = x_1, \ldots, X_N = x_N\right)$$

$$= \arg\max_{y = y_1 \ldots y_N} P\left(Y_1 = y_1, \ldots, Y_N = y_N, X_1 = x_1, \ldots, X_N = x_N\right)$$

Decoding Problem

- Decoding requires the following calculation

$$P(Y \mid X) = \frac{P(X, Y)}{P(X)}$$

$$P(Y \mid X) = \frac{P(X, Y)}{\sum_{Y} P(X, Y)}$$

Decoding Problem

- Decoding requires the following calculation

$$P(Y \mid X) = \frac{P(X, Y)}{P(X)}$$

$$P(Y \mid X) = \frac{P(X, Y)}{\sum_{Y} P(X, Y)}$$

But what is the sum in the denominator over?

Over all possible sequences $Y = y_1, \ldots, y_N$

Decoding Problem

$$\text{In general, number of sequences} = K^N$$

EXPONENTIAL IN LENGTH

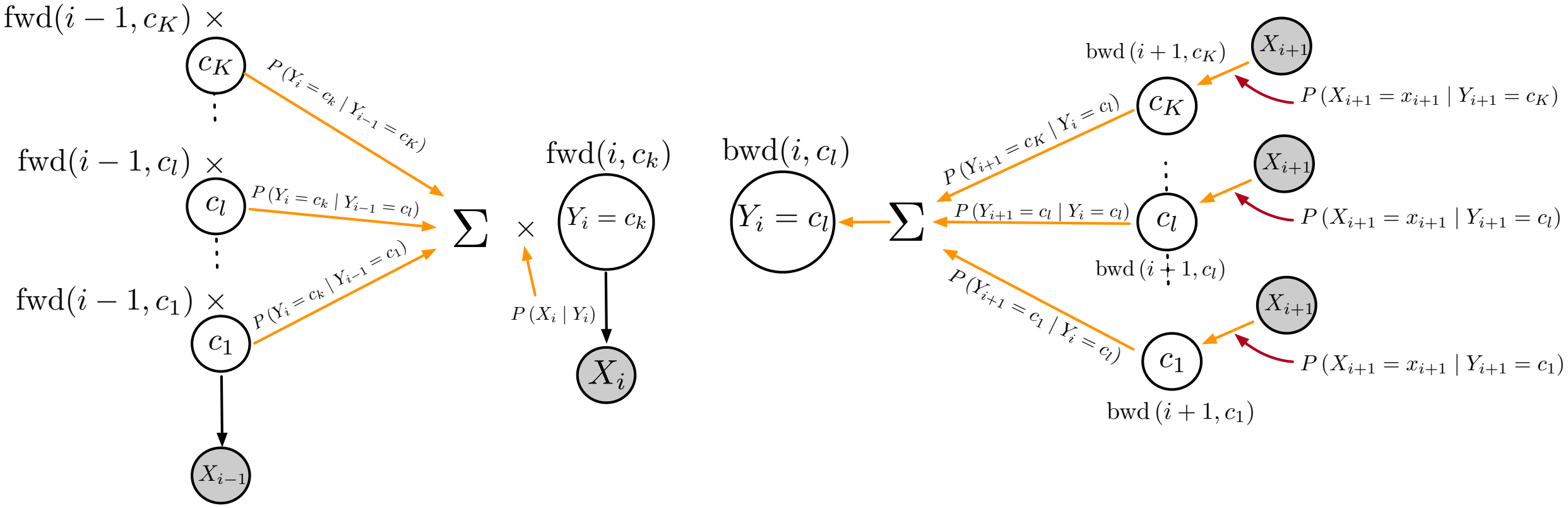
What do we do?

DYNAMIC PROGRAMMING

Forward and Backward

$$\text{forward}\,(i, c_k) := P\,(Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i)$$

$$\text{backward}(i, c_l) := P\,(X_{i+1} = x_{i+1}, \ldots, X_N = x_N \mid Y_i = c_l)$$



Forward and Backward

$$\text{forward}\,(i, c_k) := P\,(Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i)$$

$$\text{forward}\,(i, c_k) = \left(\sum_{c_l} P\,(Y_i = c_k \mid Y_{i-1} = c_l) \times \text{forward}(i-1, c_l) \right) \times P\,(X_i \mid Y_i = c_k)$$

$$\text{backward}(i, c_l) := P\,(X_{i+1} = x_{i+1}, \ldots, X_N = x_N \mid Y_i = c_l)$$

$$\text{backward}(i, c_l) = \sum_{c_k} P\,(X_{i+1} = x_{i+1} \mid Y_{i+1} = c_k) \times \text{backward}(i+1, c_k) \times P\,(Y_{i+1} = c_k \mid Y_i = c_l)$$

$$P(X) = \text{forward}(N+1, STOP)$$

$$\text{backward}(0, START) = P(X)$$

Decoding Problem

$$\text{Brute force approach } \sim O(K^N)$$

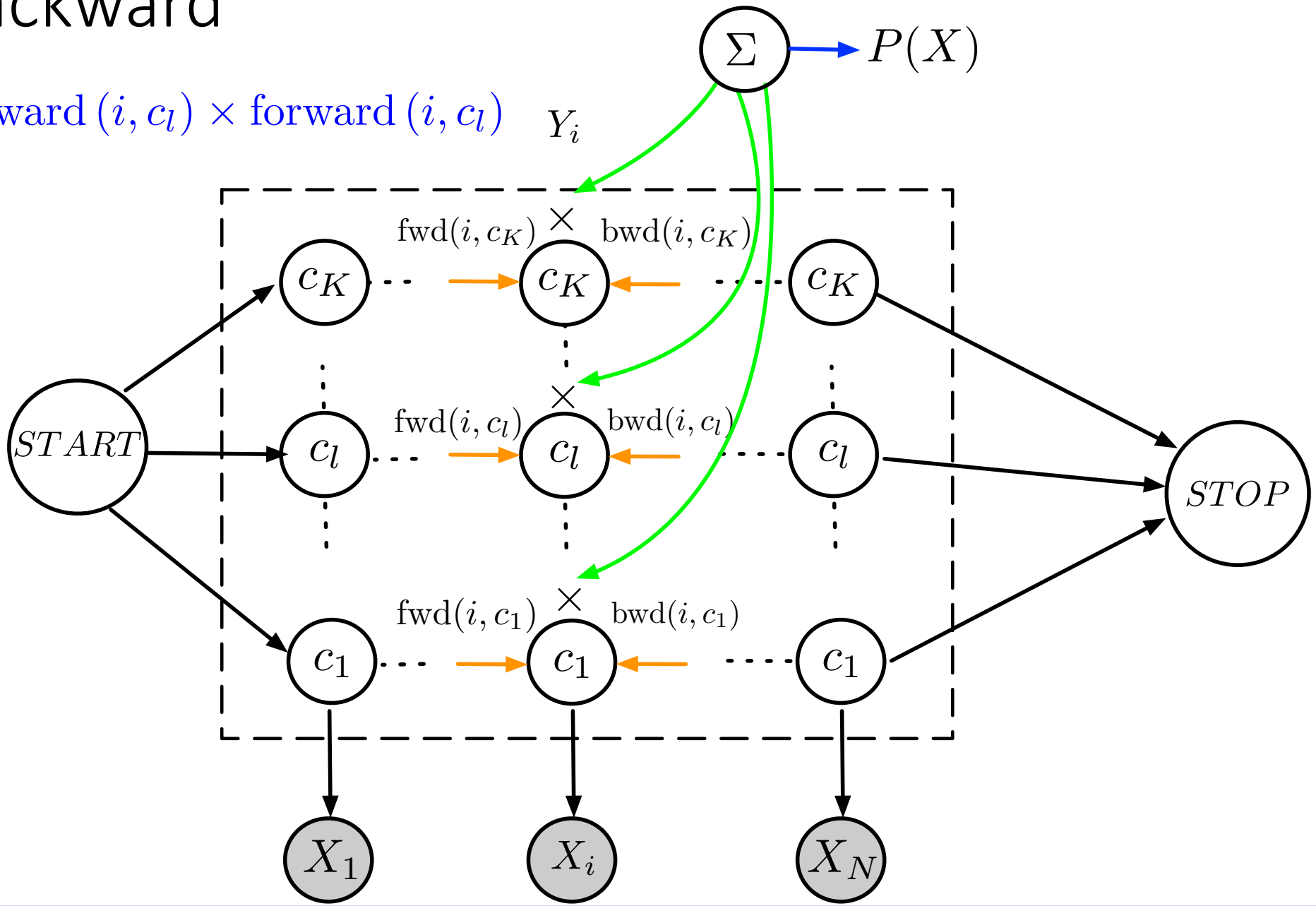
$$\text{What is the complexity of forward probability calculations?}$$

$$\text{Complexity of forward probability calculations} \sim O(NK^2)$$

We have drastically reduced number of operations:
exponential → linear

Forward Backward

$$P(X) = \sum_{c_l} \text{backward}\,(i, c_l) \times \text{forward}\,(i, c_l)$$



Decoding Problem

- Two ways possible
  - **Posterior Decoding**: Maximize probability of each state

$$y_i^* = \arg\max_{c_l} P\left(Y_i = c_l | X_1 = x_1, \ldots, X_N = x_N\right)$$

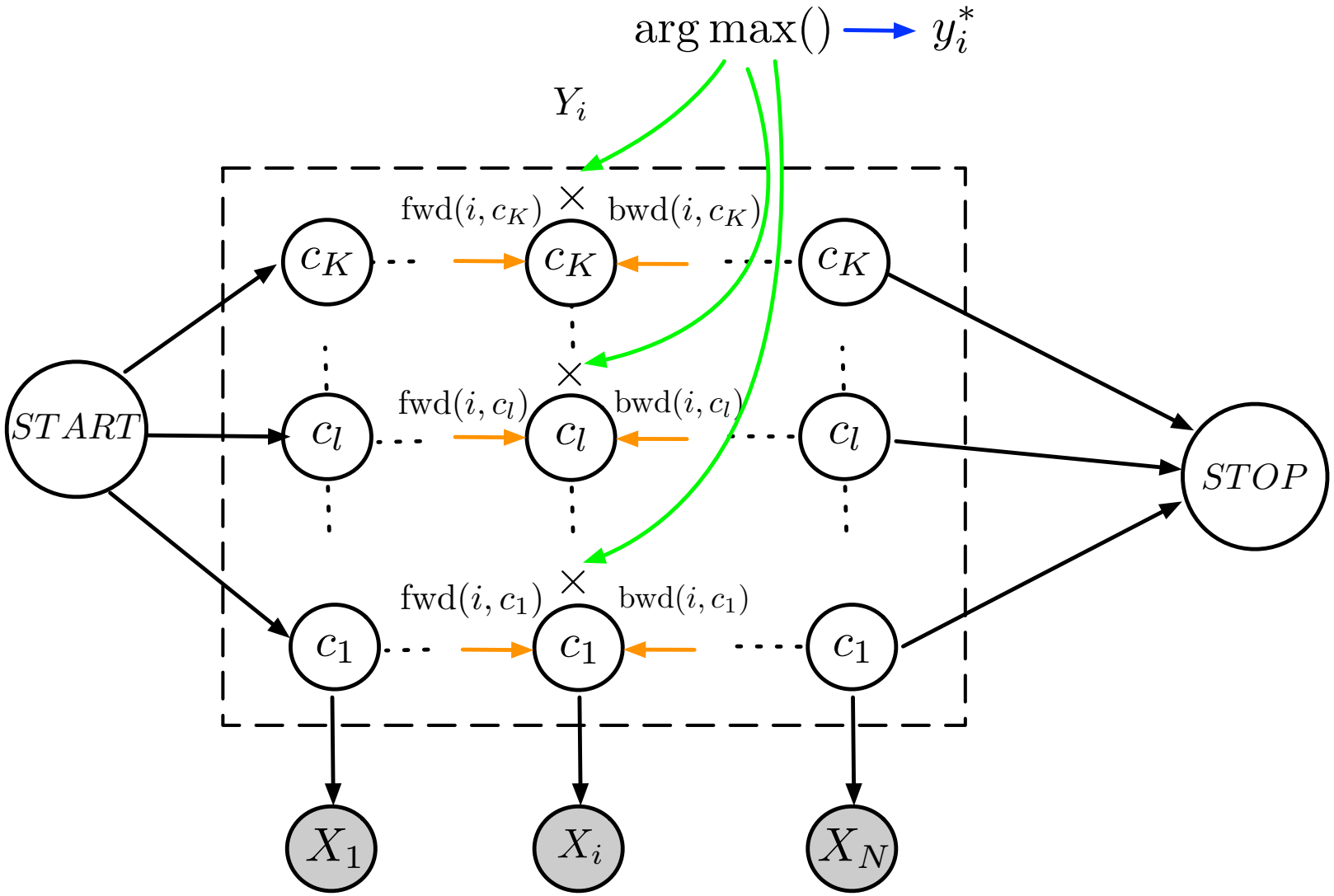
  - **Viterbi Decoding**: Maximize the probability of the entire sequence

$$y^* = \arg\max_{y=y_1 \ldots y_N} P\left(Y_1 = y_1, \ldots, Y_N = y_N | X_1 = x_1, \ldots, X_N = x_N\right)$$

$$= \arg\max_{y=y_1 \ldots y_N} P\left(Y_1 = y_1, \ldots, Y_N = y_N, X_1 = x_1, \ldots, X_N = x_N\right)$$

Posterior Decoding

$$y_i^* = \arg\max_{c_l} \text{forward}(i, c_l) \times \text{backward}(i, c_l)$$



Decoding Problem

- Two ways possible
  - **Posterior Decoding**: Maximize probability of each state

$$y_i^* = \arg\max_{c_l} P\left(Y_i = c_l | X_1 = x_1, \ldots, X_N = x_N\right)$$

  - **Viterbi Decoding**: Maximize the probability of the entire sequence

$$y^* = \arg\max_{y=y_1 \ldots y_N} P\left(Y_1 = y_1, \ldots, Y_N = y_N | X_1 = x_1, \ldots, X_N = x_N\right)$$

$$= \arg\max_{y=y_1 \ldots y_N} P\left(Y_1 = y_1, \ldots, Y_N = y_N, X_1 = x_1, \ldots, X_N = x_N\right)$$

Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1, \ldots, y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

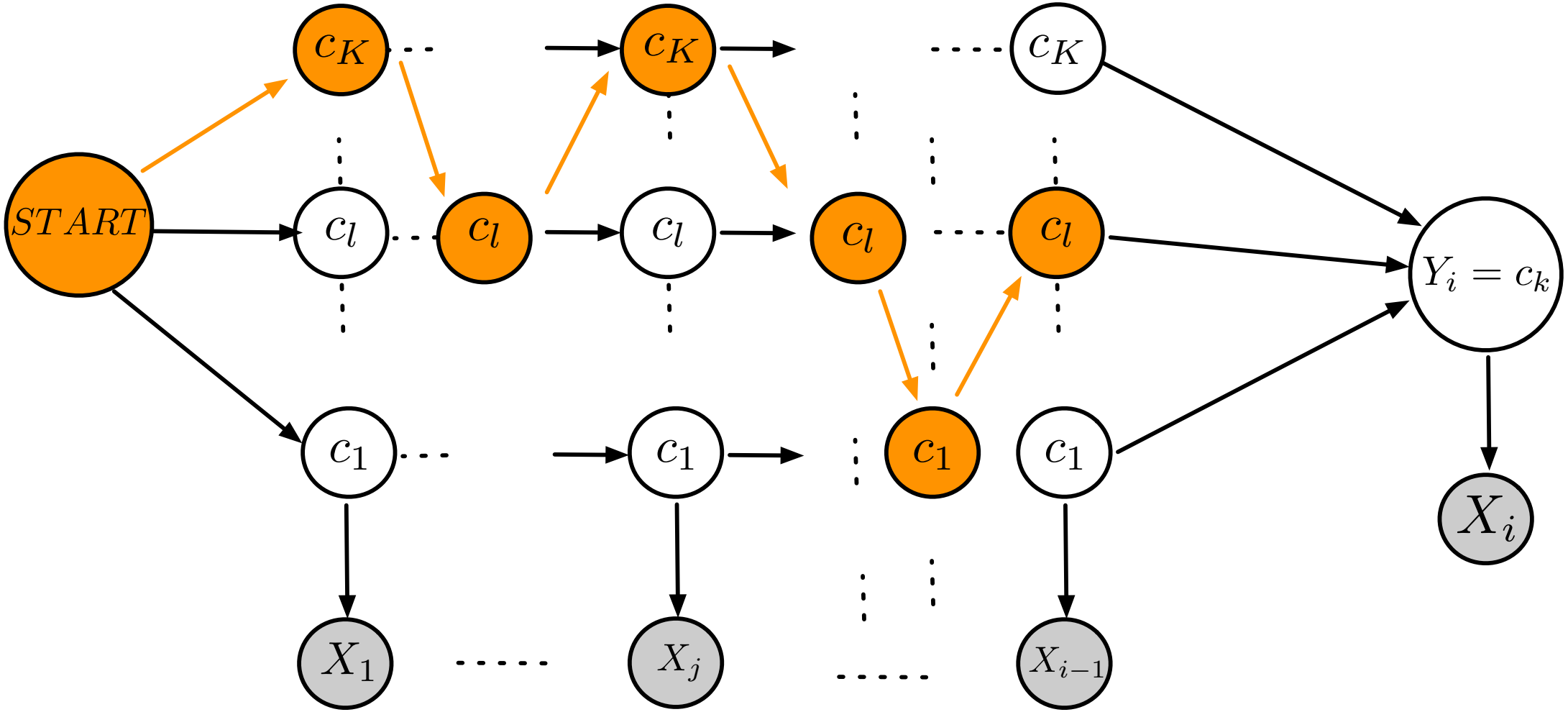
Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1,\ldots,y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

The Viterbi trellis represents the path with maximum probability at position/step *i* when we are in state $Y_i = y_i$ and that we have observed $x_1, \ldots, x_i$ up to that position.

Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1,\ldots,y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$



Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1, \ldots, y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1, \ldots, y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

$$\text{viterbi}(i, c_k) = \left(\max_{c_l} \text{viterbi}(i-1, c_l) \times P\left(Y_i = c_k \mid Y_{i-1} = c_l\right) \right) \times P\left(X_i \mid Y_i = c_k\right)$$

Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1, \ldots, y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

$$= \max_{y_1, \ldots, y_{i-1}} P\left(X_i \mid Y_i = c_k, Y_1, \ldots, Y_{i-1}, X_1, \ldots, X_{i-1}\right) \times P\left(Y_1, \ldots, Y_i = c_k, X_1, \ldots, X_{i-1}\right)$$

Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1,\ldots,y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k, Y_1, \ldots, Y_{i-1}, X_1, \ldots, X_{i-1}\right) \times P\left(Y_1, \ldots, Y_i = c_k, X_1, \ldots, X_{i-1}\right)$$

$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k\right) \times P\left(Y_i = c_k \mid Y_1, .., Y_{i-1}, X_1, .., X_{i-1}\right) \times P\left(Y_1, .., Y_{i-1}, X_1, .., X_{i-1}\right)$$

Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1,\ldots,y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k, Y_1, \ldots, Y_{i-1}, X_1, \ldots, X_{i-1}\right) \times P\left(Y_1, \ldots, Y_i = c_k, X_1, \ldots, X_{i-1}\right)$$

$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k\right) \times P\left(Y_i = c_k \mid Y_1, .., Y_{i-1}, X_1, .., X_{i-1}\right) \times P\left(Y_1, .., Y_{i-1}, X_1, .., X_{i-1}\right)$$

$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k\right) \times P\left(Y_i = c_k \mid Y_{i-1}\right) \times P\left(Y_1, \ldots, Y_{i-1}, X_1, \ldots, X_{i-1}\right)$$

Viterbi Decoding

$$\text{viterbi}(i, c_k) := \max_{y_1,\ldots,y_{i-1}} P\left(Y_1 = y_1, \ldots, Y_i = c_k, X_1 = x_1, \ldots, X_i = x_i\right)$$

$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k, Y_1, \ldots, Y_{i-1}, X_1, \ldots, X_{i-1}\right) \times P\left(Y_1, \ldots, Y_i = c_k, X_1, \ldots, X_{i-1}\right)$$

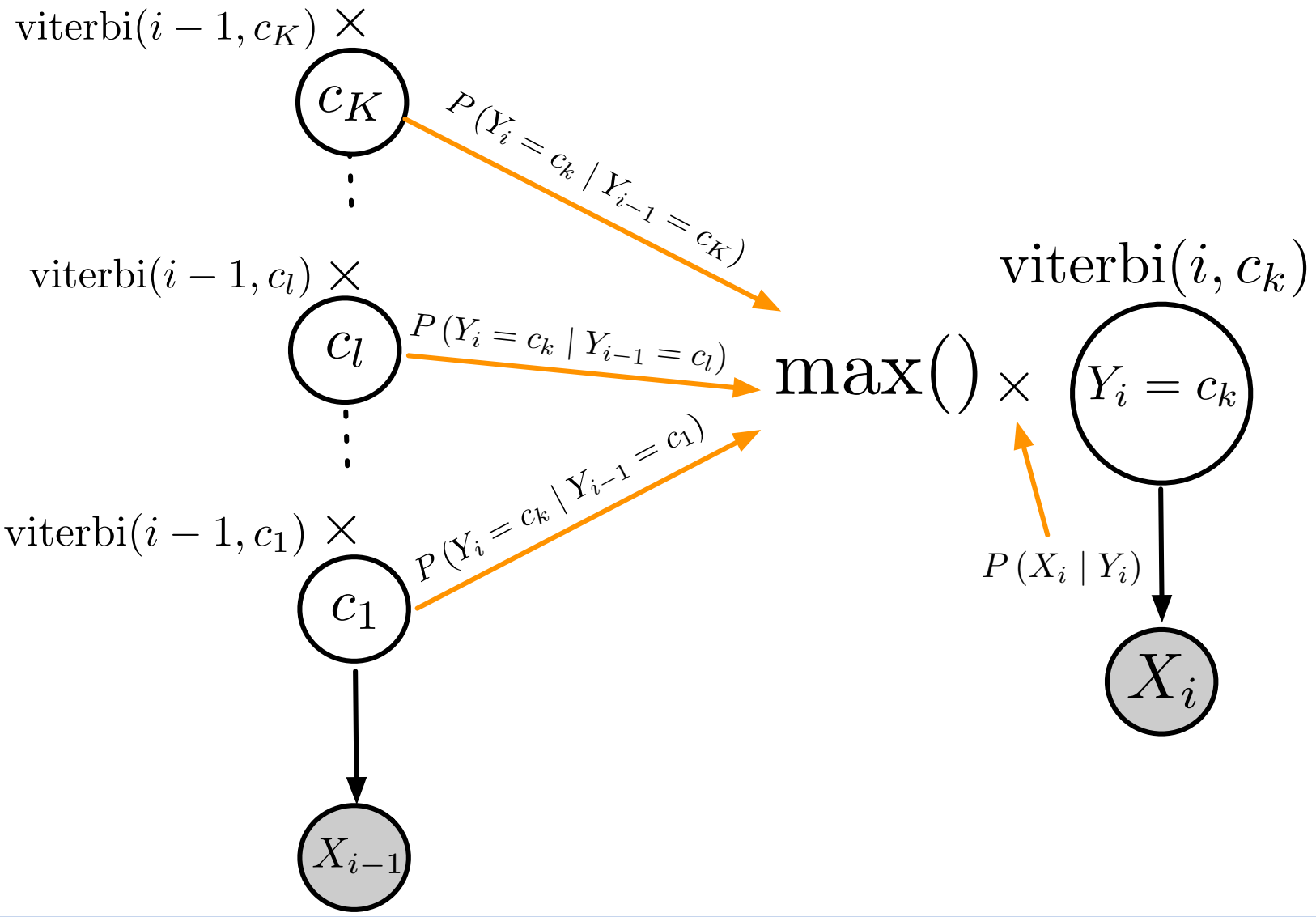
$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k\right) \times P\left(Y_i = c_k \mid Y_1, .., Y_{i-1}, X_1, .., X_{i-1}\right) \times P\left(Y_1, .., Y_{i-1}, X_1, .., X_{i-1}\right)$$

$$= \max_{y_1,\ldots,y_{i-1}} P\left(X_i \mid Y_i = c_k\right) \times P\left(Y_i = c_k \mid Y_{i-1}\right) \times P\left(Y_1, \ldots, Y_{i-1}, X_1, \ldots, X_{i-1}\right)$$

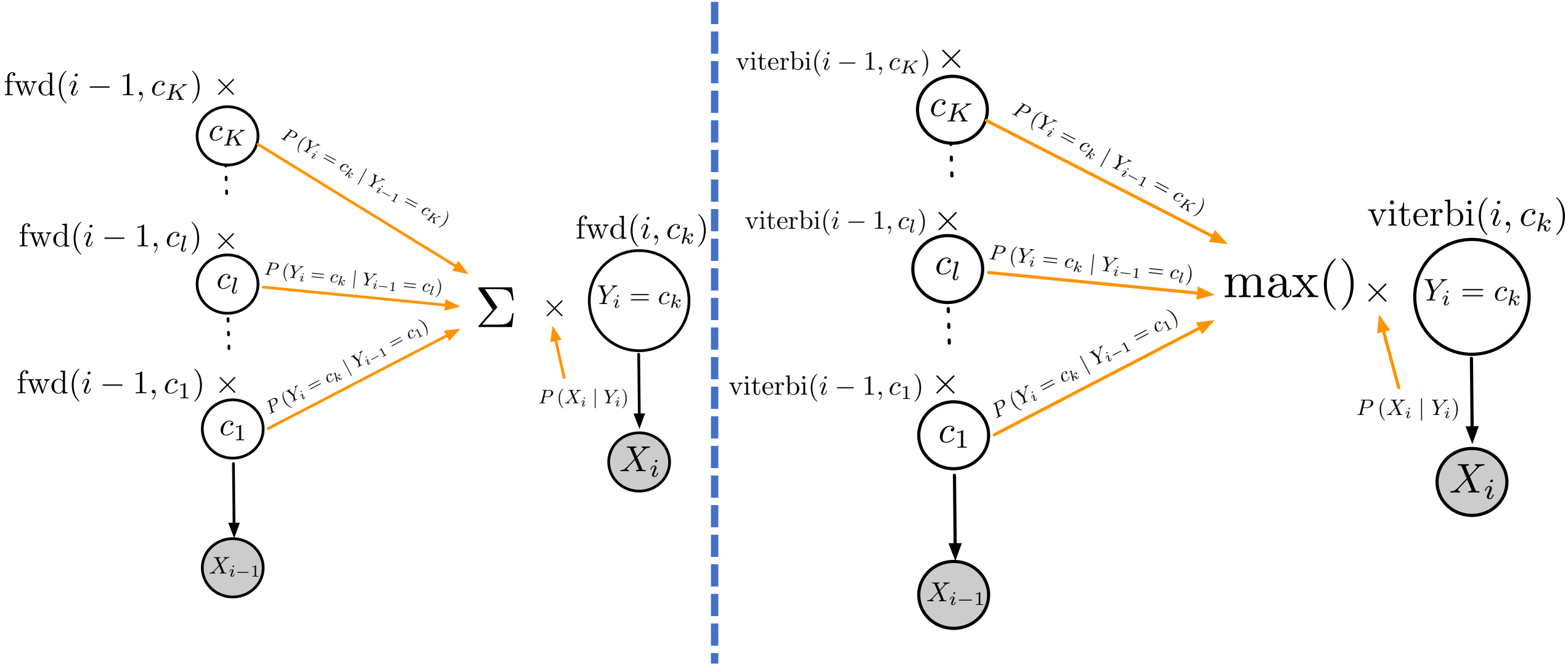
$$= \left(\max_{c_l} \text{viterbi}(i-1, c_l) \times P\left(Y_i = c_k \mid Y_{i-1} = c_l\right)\right) \times P\left(X_i \mid Y_i = c_k\right)$$

$$\text{viterbi}(i, c_k) = \left(\max_{c_l} \text{viterbi}(i-1, c_l) \times P\left(Y_i = c_k \mid Y_{i-1} = c_l\right) \right) \times P\left(X_i \mid Y_i = c_k\right)$$

Viterbi Decoding



Forward vs Viterbi



Viterbi Decoding

The Viterbi trellis represents the path with maximum probability in position i when we are in state $Y_i = y_i$ and that we have observed $x_1, \ldots, x_i$ up to that position.

But once we reach the last state how do we get the most likely path?

Viterbi Decoding

The Viterbi trellis represents the path with maximum probability in position i when we are in state $Y_i = y_i$ and that we have observed $x_1, \ldots, x_i$ up to that position.

But once we reach the last state how do we get the most likely path?

Back-Tracking

Viterbi Decoding

The Viterbi trellis represents the path with maximum probability in position *i* when we are in state $Y_i = y_i$ and that we have observed $x_1, \ldots, x_i$ up to that position.

But once we reach the last state how do we get the most likely path?

Back-Tracking

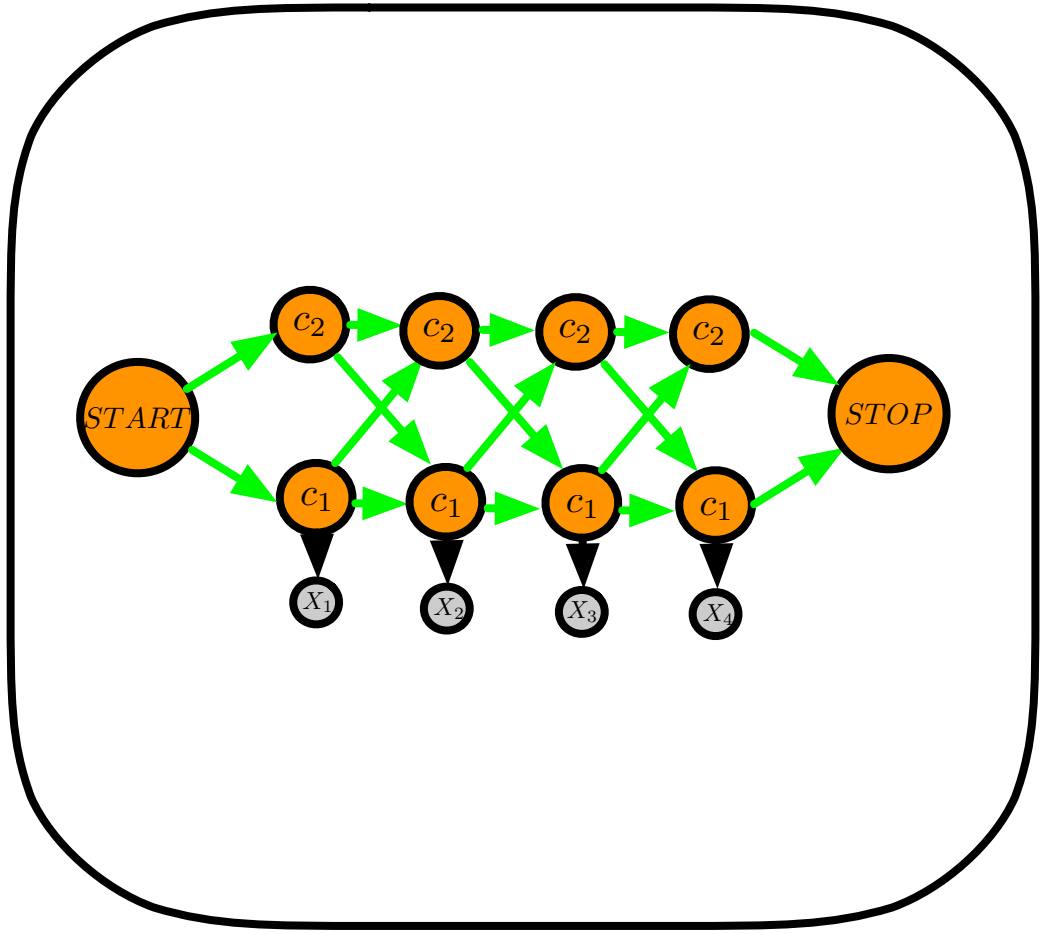
$$\text{backtrack}(i, c_k) = \left(\arg\max_{c_l} \text{viterbi}(i-1, c_l) \times P\left(Y_i = c_k \mid Y_{i-1} = c_l\right) \right)$$

How do we get most probable sequence?

$\{X_1, X_2, \ldots, X_N\}$

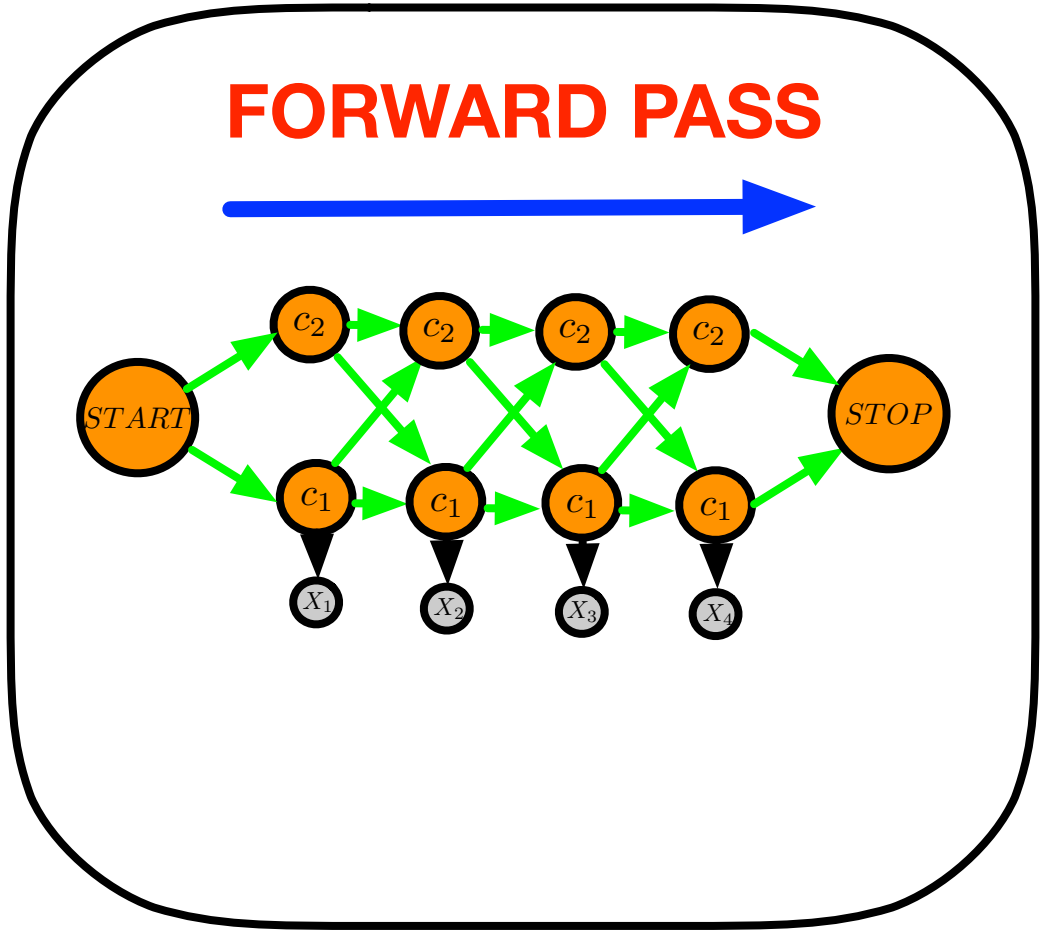
How do we get most probable sequence?

$\{X_1, X_2, \ldots, X_N\}$



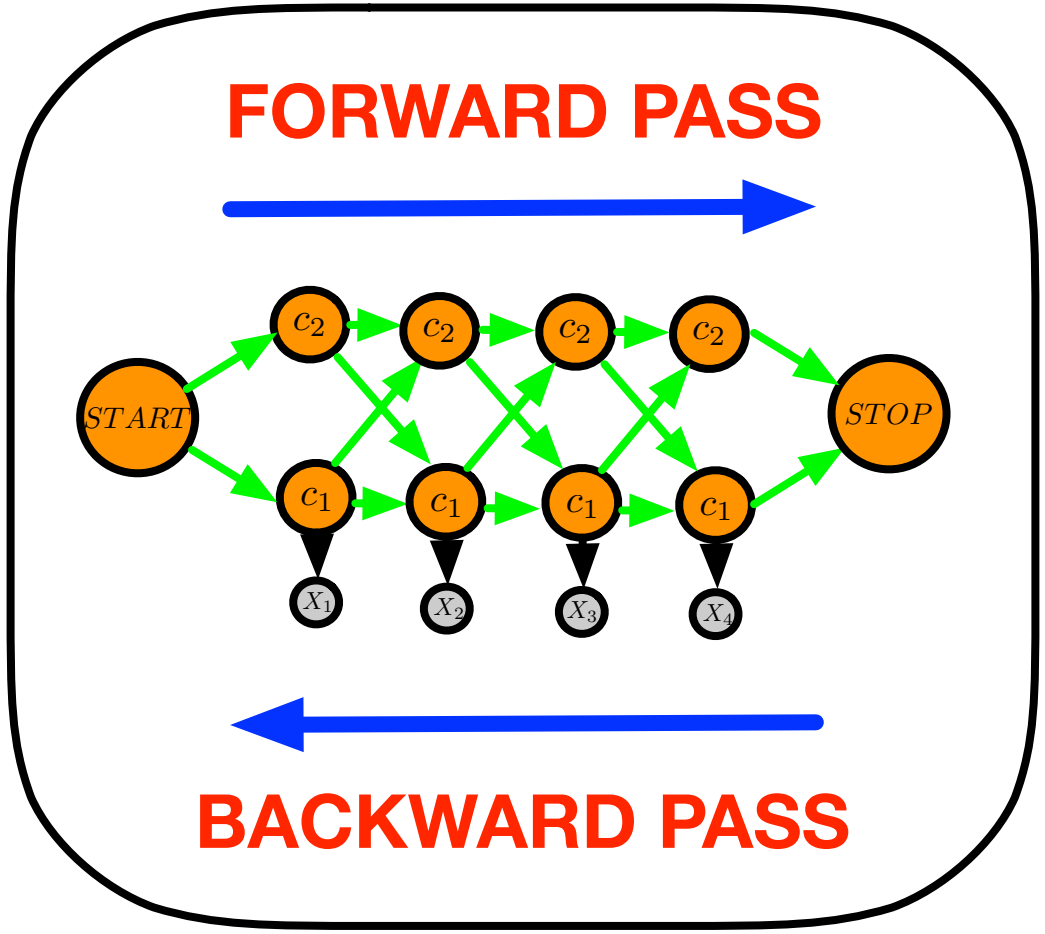
How do we get most probable sequence?

$\{X_1, X_2, \ldots, X_N\}$



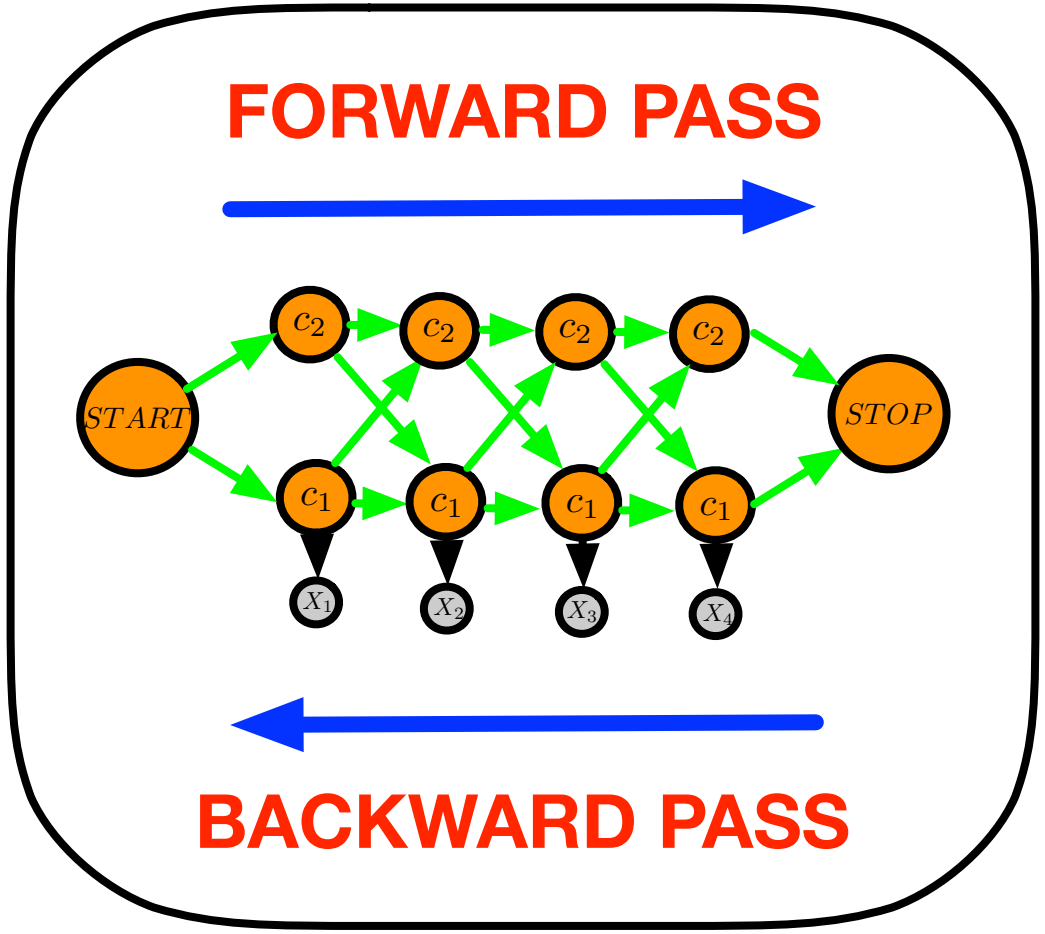
How do we get most probable sequence?

$\{X_1, X_2, \ldots, X_N\}$



How do we get most probable sequence?

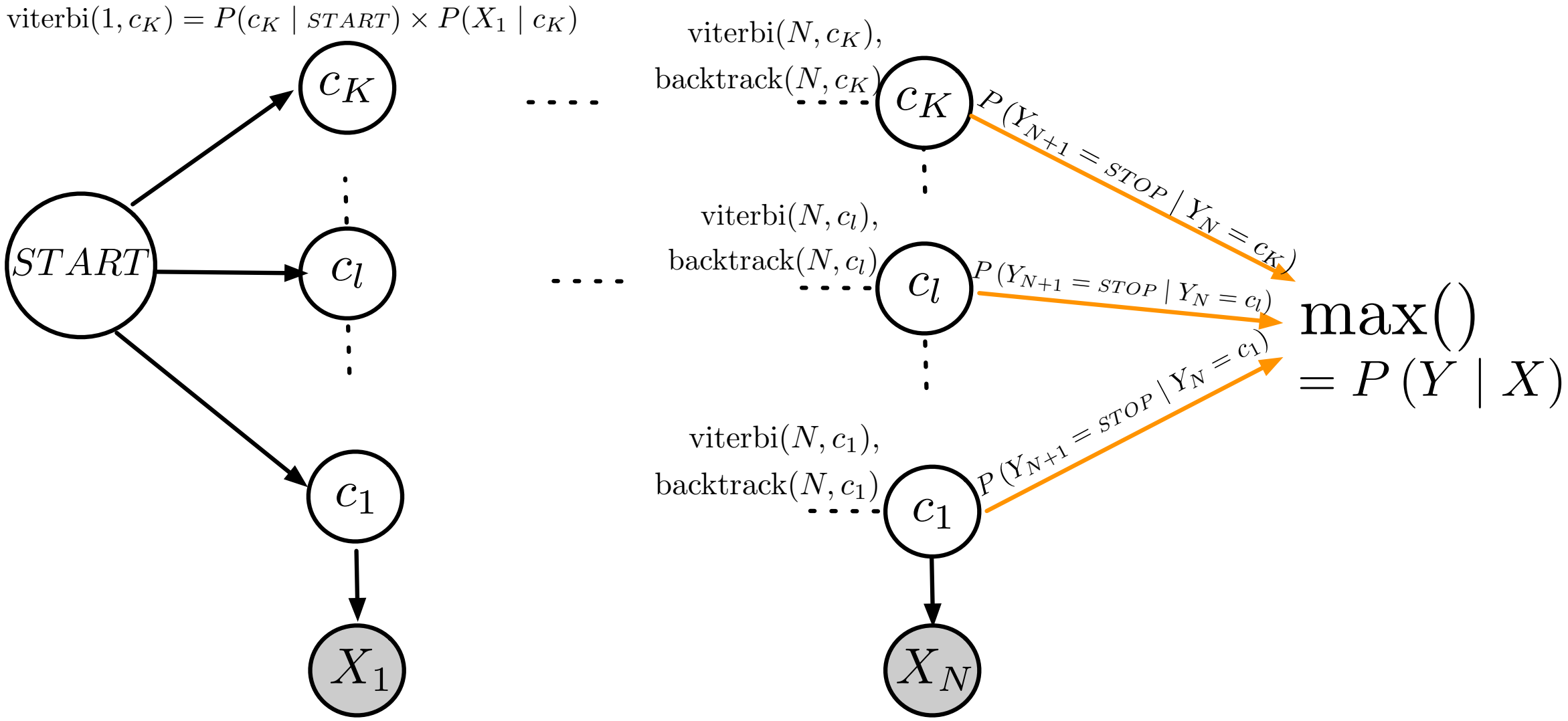
$\{X_1, X_2, \ldots, X_N\}$



$\{Y_1, Y_2, \ldots, Y_N\}$

How do we get most probable sequence?

Forward Pass



How do we get most probable sequence?

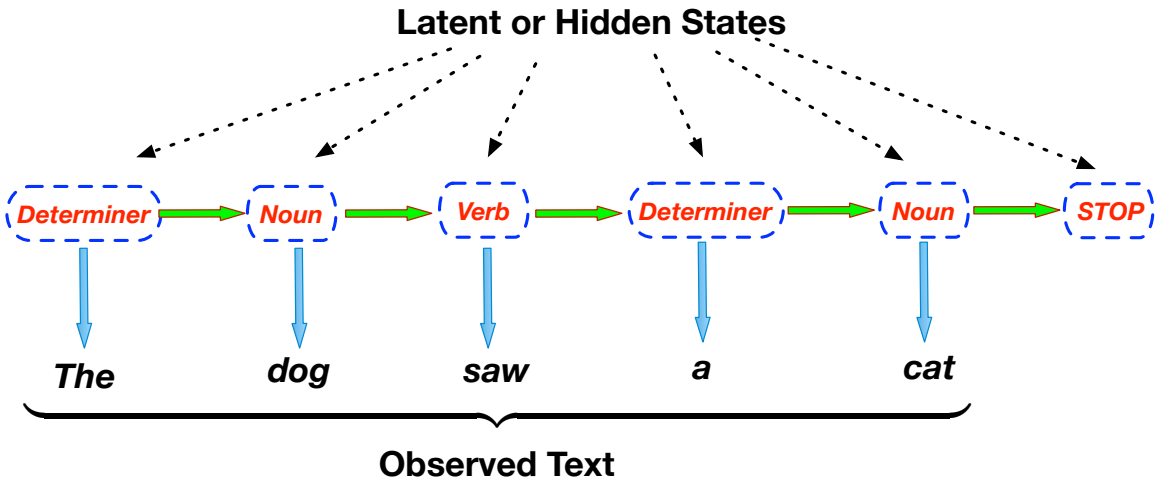
Backward Pass

$$\hat{y}_N = \arg\max_{c_l} P(STOP \mid c_l) \times \text{viterbi}(N, c_l)$$

$$\text{for } i = (N-1) \ldots 1 :$$

$$\hat{y}_i = \text{backtrack}(i+1, \hat{y}_{i+1})$$

Hidden Markov Models (HMM)



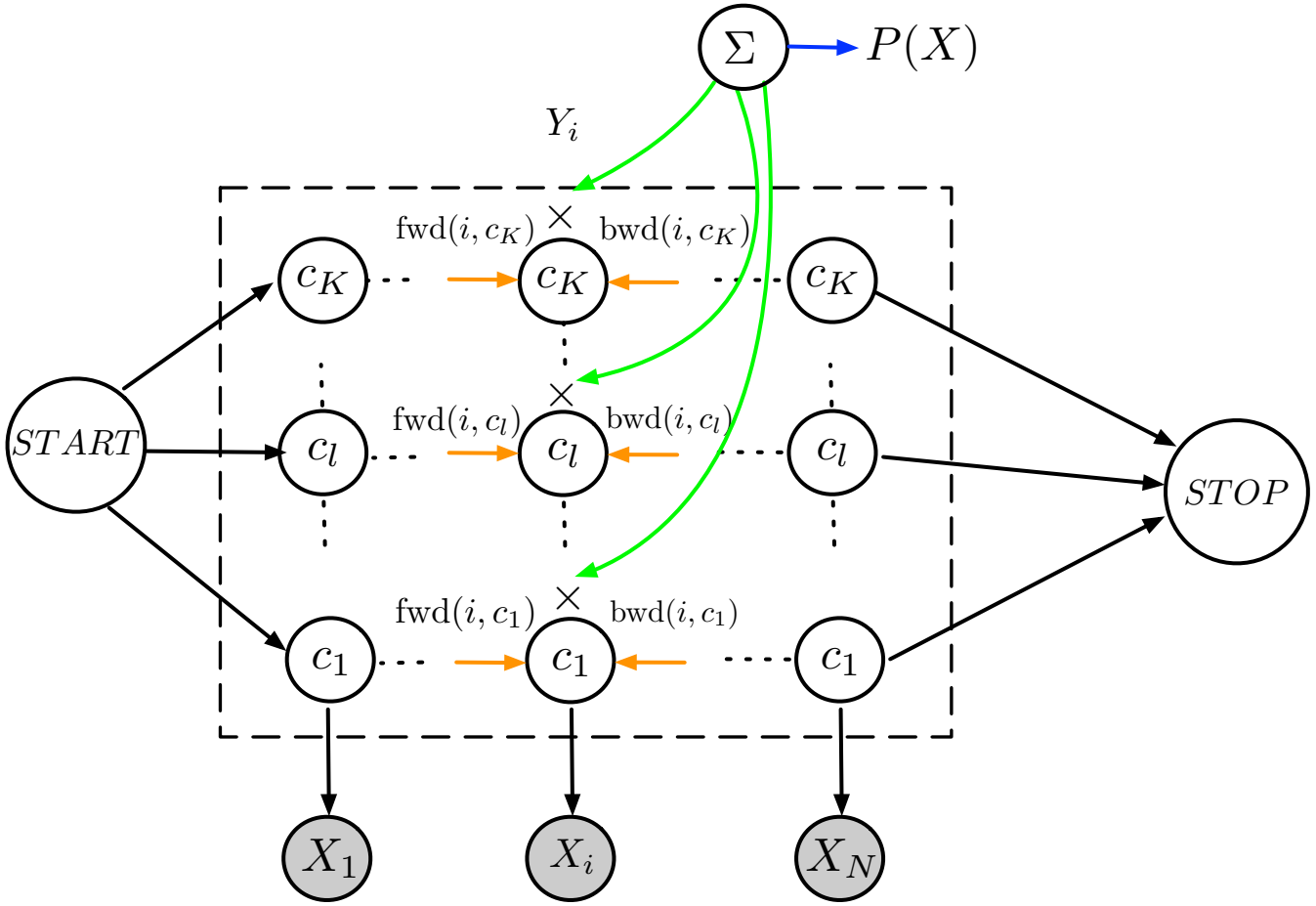
Three Questions:

1. ***Learning Problem***: How do we estimate the parameters of distributions?
2. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?
3. ***Likelihood Problem***: Given the parameters of the distributions what is the probability of the observed sequence?

$$P(X) = ?$$

Observation Probability

$$P(X) = \sum_{c_l} \text{backward}\,(i, c_l) \times \text{forward}\,(i, c_l)$$



HMM Summary

Learning Problem: How do we estimate the parameters of distributions?

Counting

Decoding Problem: Given the observed text what is the hidden POS sequence that best explains the observation?

Forward Backward
Viterbi

Likelihood Problem: Given the parameters of the distributions what is the probability of the observed sequence?

Forward Backward

Unsupervised Sequence Modeling

- We saw supervised setting where we are given paired observations and states
- What if we were given only observations? → Practical Scenario
- Can we infer states (POS Tags)?

Unsupervised Sequence Modeling

- ***Supervised Setting***: Given set of paired observations and state sequences

$$\mathcal{D}_L = \{(x^1, y^1), ..., (x^M, y^M)\}$$

$$x^i = X_1^i \; X_2^i .... \; X_N^i$$

$$y^i = Y_1^i \; Y_2^i .... \; Y_N^i$$

- ***Unsupervised Setting***: Given set of observations only

$$\mathcal{D}_\mathcal{U} = \{x^1, \cdots, x^M\}$$

$$x^i = X_1^i \; X_2^i \; \cdots \; X_N^i$$

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

$$\mathcal{NLL}_\theta = -\frac{1}{M}\sum_{m=1}^{M} \log P_\theta(X)$$

$$\theta = \text{Parameters (transition and emission probabilities)}$$

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

$$\mathcal{NLL}_\theta = -\frac{1}{M}\sum_{m=1}^{M} \log P_\theta(X)$$

$$= -\frac{1}{M}\sum_{m=1}^{M} \log \sum_{y^m \in \Lambda} P_\theta(X = x^m, Y = y^m)$$

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Expectation Maximization (EM) Algorithm

(Baum-Welch Algorithm)

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Expectation Maximization (EM) Algorithm

(Baum-Welch Algorithm)

(Forward-Backward Algorithm)

Intuition:

1. If we had observed hidden states for all sentences in the corpus then we could have easily computed the likelihood estimates of parameters

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Expectation Maximization (EM) Algorithm

(Baum-Welch Algorithm)

(Forward-Backward Algorithm)

Intuition:

1. If we had observed hidden states for all sentences in the corpus then we could have easily computed the likelihood estimates of parameters
2. On the other hand if we had the model parameters, we could label the data

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Expectation Maximization (EM) Algorithm

(Baum-Welch Algorithm)

(Forward-Backward Algorithm)

Intuition:

1. If we had observed hidden states for all sentences in the corpus then we could have easily computed the likelihood estimates of parameters
2. On the other hand if we had the model parameters, we could label the data
3. Given the observation data, we need to collect expected counts that represents number of times that each hidden variable is expected to be used

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Expectation Maximization (EM) Algorithm

(Baum-Welch Algorithm)

(Forward-Backward Algorithm)

Intuition:

1. If we had observed hidden states for all sentences in the corpus then we could have easily computed the likelihood estimates of parameters
2. On the other hand if we had the model parameters, we could label the data
3. Given the observation data, we need to collect expected counts that represents number of times that each hidden variable is expected to be used.
4. These expected counts can be used during learning as fake observations of hidden states.

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Expectation Maximization (EM) Algorithm

(Baum-Welch Algorithm)

(Forward-Backward Algorithm)

The key idea:

Randomly initialize parameters (transition and emission probabilities).

Iterate:

E-Step: Use forward-backward algorithm to calculate the expected posterior transition probability distribution and expected posterior emission probability distribution.

Unsupervised Sequence Modeling

Given only observations, can we infer POS Tags?

Expectation Maximization (EM) Algorithm

(Baum-Welch Algorithm)

(Forward-Backward Algorithm)

The key idea:

Randomly initialize parameters (transition and emission probabilities).

Iterate:

E-Step: Use forward-backward algorithm to calculate the expected posterior transition probability distribution and expected posterior emission probability distribution.

M-Step: Use the expected posteriors to update the transition and emission probabilities

Posteriors

Calculate Posterior distributions

$$P_\theta(Y_{t+1}, Y_t \mid x^m = X_{1:N})$$

$$P_\theta(Y_t \mid x^m = X_{1:N})$$

Posteriors

Calculate Posterior distributions

$$P_\theta(Y_{t+1}, Y_t \mid x^m = X_{1:N})$$

$$P(Y_{t+1} = c_j, Y_t = c_i \mid X_{1:N}) = \frac{P(Y_{t+1} = c_j, Y_t = c_i, X_{1:N})}{P(X_{1:N})}$$

Posteriors

Calculate Posterior distributions

$$P_\theta(Y_{t+1}, Y_t \mid x^m = X_{1:N})$$

$$P(Y_{t+1} = c_j, Y_t = c_i \mid X_{1:N}) = \frac{P(Y_{t+1} = c_j, Y_t = c_i, X_{1:N})}{P(X_{1:N})}$$

$$= \frac{\text{fwd}(t, c_i) \times P(Y_{t+1} = c_j \mid Y_t = c_i) \times P(X_{t+1} \mid Y_{t+1} = c_j) \times \text{bwd}(t+1, c_j)}{P(X_{1:N})}$$

Posteriors

Calculate Posterior distributions

$$P_\theta(Y_{t+1}, Y_t \mid x^m = X_{1:N})$$

$$P(Y_{t+1} = c_j, Y_t = c_i \mid X_{1:N}) = \frac{P(Y_{t+1} = c_j, Y_t = c_i, X_{1:N})}{P(X_{1:N})}$$

$$= \frac{\text{fwd}(t, c_i) \times P(Y_{t+1} = c_j \mid Y_t = c_i) \times P(X_{t+1} \mid Y_{t+1} = c_j) \times \text{bwd}(t+1, c_j)}{P(X_{1:N})}$$

$$= \frac{\text{fwd}(t, c_i) \times P(Y_{t+1} = c_j \mid Y_t = c_i) \times P(X_{t+1} \mid Y_{t+1} = c_j) \times \text{bwd}(t+1, c_j)}{\sum_{c_l} \text{fwd}\,(t, c_l) \times \text{bwd}\,(t, c_l)}$$

$$= \xi_t(c_i, c_j)$$

Posteriors

Calculate Posterior distributions

$$P_\theta(Y_t \mid x^m = X_{1:N})$$

$$P_\theta(Y_t = c_j \mid X_{1:N}) = \frac{P(Y_t = c_j, X_{1:N})}{P(X_{1:N})}$$

Posteriors

Calculate Posterior distributions

$$P_\theta(Y_t \mid x^m = X_{1:N})$$

$$P_\theta(Y_t = c_j \mid X_{1:N}) = \frac{P(Y_t = c_j, X_{1:N})}{P(X_{1:N})}$$

$$= \frac{\text{fwd}(t, c_j) \times \text{bwd}(t, c_j)}{\sum_{c_l} \text{bwd}\,(t, c_l) \times \text{fwd}\,(t, c_l)}$$

$$= \gamma_t(c_j)$$

Expected Counts

Calculate Expected Counts

$$\hat{C}_{init}(c_k) = \sum_{m=1}^{M} P_\theta(Y_1^m = c_k \mid x^m = X_{1:N})$$

$$\hat{C}_{trans}(c_k, c_l) = \sum_{m=1}^{M} \sum_{t=2}^{N} P_\theta(Y_{t+1}^m = c_k, Y_t^m = c_l \mid x^m = X_{1:N})$$

$$\hat{C}_{final}(c_k) = \sum_{m=1}^{M} P_\theta(Y_N^m = c_k \mid x^m = X_{1:N})$$

$$\hat{C}_{emiss}(c_k, w_j) = \sum_{m=1}^{M} \sum_{t=1}^{N} \delta_{(x_t^m = w_j)} P_\theta(Y_t^m = c_k \mid x^m = X_{1:N})$$

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

for $i = 1$ to T do

end for

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

for $i = 1$ to T do

E-Step:

end for

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

for $i = 1$ to T do

E-Step:

Clear counts: $\hat{C}_{init}(.) = \hat{C}_{trans}(.,.) = \hat{C}_{final}(.) = \hat{C}_{emiss}(.,.) = 0$

end for

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

for $i = 1$ to T do

E-Step:

Clear counts: $\hat{C}_{init}(.) = \hat{C}_{trans}(.,.) = \hat{C}_{final}(.) = \hat{C}_{emiss}(.,.) = 0$

for $x^m \in \mathcal{D}_{\mathcal{U}}$ do

end for

end for

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

for $i = 1$ to T do

E-Step:

Clear counts: $\hat{C}_{init}(.) = \hat{C}_{trans}(.,.) = \hat{C}_{final}(.) = \hat{C}_{emiss}(.,.) = 0$

for $x^m \in \mathcal{D}_{\mathcal{U}}$ do

Calculate posterior expectations $P_\theta(Y_{t+1}, Y_t \mid x^m = X_{1:N}), P_\theta(Y_t \mid x^m = X_{1:N})$ using current θ

Update expected counts

end for

end for

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

for $i = 1$ to T do

E-Step:

Clear counts: $\hat{C}_{init}(.) = \hat{C}_{trans}(.,.) = \hat{C}_{final}(.) = \hat{C}_{emiss}(.,.) = 0$

for $x^m \in \mathcal{D}_{\mathcal{U}}$ do

Calculate posterior expectations $P_\theta(Y_{t+1}, Y_t \mid x^m = X_{1:N}), P_\theta(Y_t \mid x^m = X_{1:N})$ using current θ

Update expected counts

end for

M-Step:

end for

EM Algorithm

Input: $\mathcal{D}_{\mathcal{U}} = \{x^1, \cdots, x^M\}$

Randomly initialize parameters θ

for $i = 1$ to T do

E-Step:

Clear counts: $\hat{C}_{init}(.) = \hat{C}_{trans}(.,.) = \hat{C}_{final}(.) = \hat{C}_{emiss}(.,.) = 0$

for $x^m \in \mathcal{D}_{\mathcal{U}}$ do

Calculate posterior expectations $P_\theta(Y_{t+1}, Y_t \mid x^m = X_{1:N}), P_\theta(Y_t \mid x^m = X_{1:N})$ using current θ

Update expected counts

end for

M-Step:

Update parameters θ based on counts

end for

HMM Summary

Supervised Setting:

Learning Problem: How do we estimate the parameters of distributions?

Counting

Decoding Problem: Given the observed text what is the hidden POS sequence that best explains the observation?

Forward Backward

Viterbi

Likelihood Problem: Given the parameters of the distributions what is the probability of the observed sequence?

Forward Backward

Unsupervised Setting:

Expectation Maximization

QUIZ

- Suppose you are given data about the activities that some person did over different days.
- We know that the activity that a person does on a given day is dependent on the weather on that day.
- We would like to predict the most likely weather pattern by observing the activity pattern.
- Just to simplify our problem, let us assume:
  - We have only 3 different activities: STUDY, SHOP, PLAY
  - We have only 2 possible weather conditions: SUNNY, RAIN

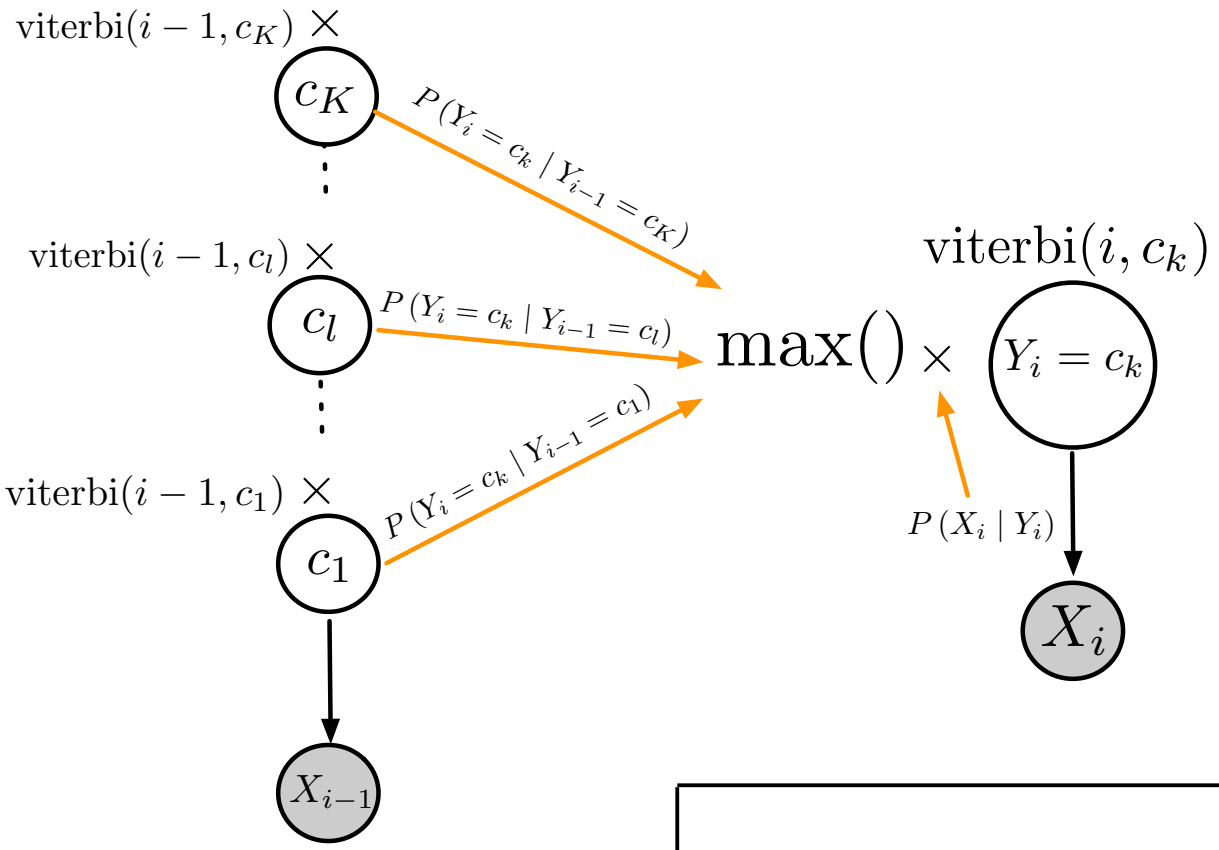
QUIZ

	STUDY (x_1)	SHOP (x_2)	PLAY (x_3)
SUNNY (c_1)	0.2	0.4	0.4
RAIN (c_2)	0.5	0.4	0.1

	SUNNY (c_1)	RAIN (c_2)
SUNNY (c_1)	0.6	0.4
RAIN (c_2)	0.5	0.5

	SUNNY (c_1)	RAIN (c_2)
START	0.8	0.2

What is the most probable sequence for the observation PLAY, STUDY, PLAY?



	STUDY (x_1)	SHOP (x_2)	PLAY (x_3)
SUNNY (c_1)	0.2	0.4	0.4
RAIN (c_2)	0.5	0.4	0.1

	SUNNY (c_1)	RAIN (c_2)
SUNNY (c_1)	0.6	0.4
RAIN (c_2)	0.5	0.5

	SUNNY (c_1)	RAIN (c_2)
START	0.8	0.2

What is the most probable sequence for the observation PLAY, STUDY, PLAY?

	SUNNY (c_1)	RAIN (c_2)
viterbi(1,c_l)	?	?
viterbi(2,c_l)	?	?
viterbi(3,c_l)	?	?

References

1. Michael Collin's NLP Lecture Notes: http://www.cs.columbia.edu/~mcollins/hmms-spring2013.pdf
2. Chapter 6, Speech and Language Processing, Dan Jurafsky and James Martin
3. LxMLS Lab Guide: http://lxmls.it.pt/2016/LxMLS2016.pdf
4. JasonEisner. An interactive spreadsheet for teaching the forward-backward algorithm. https://tinyurl.com/rm3qq7v