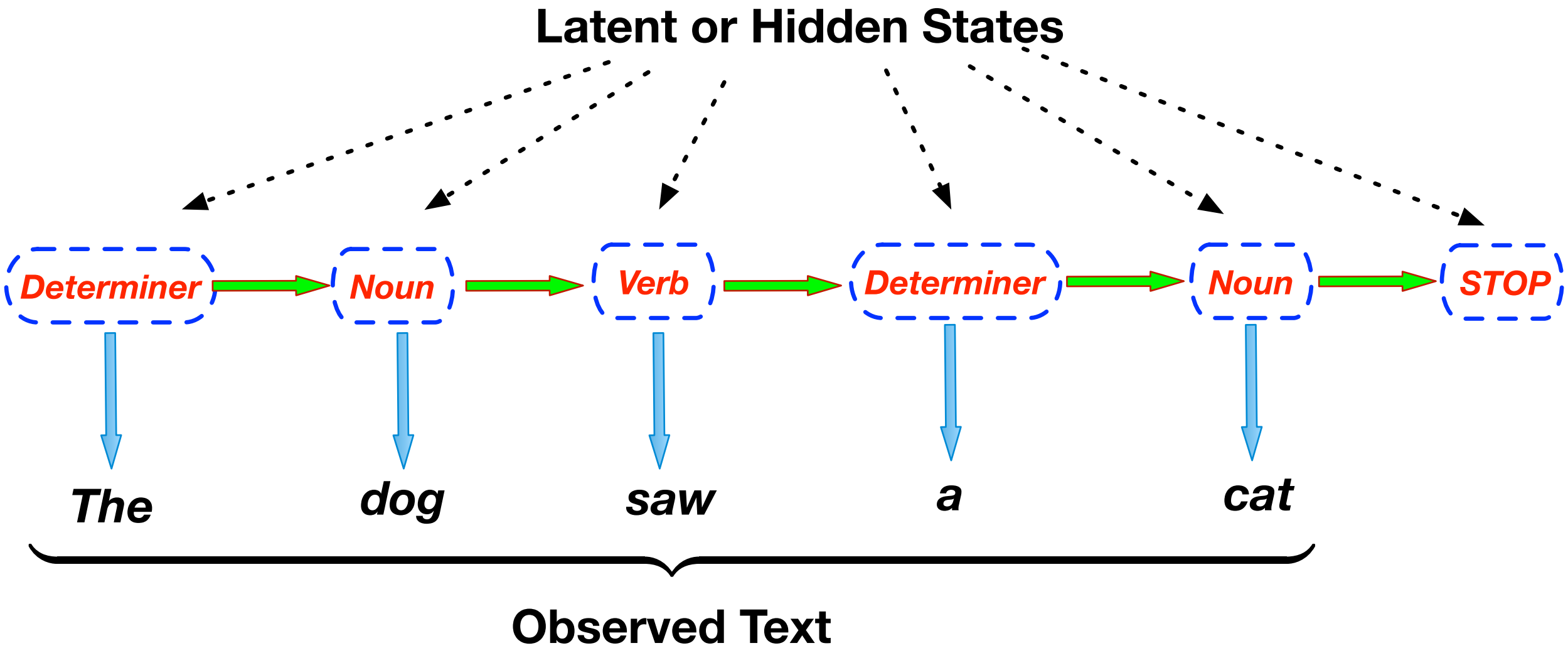
Special Topics in Natural Language Processing

CS698O

Ashutosh Modi
CSE Department, IIT Kanpur

Lecture 13: Sequence Prediction 4
Feb 3, 2020

Hidden Markov Models (HMM)



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

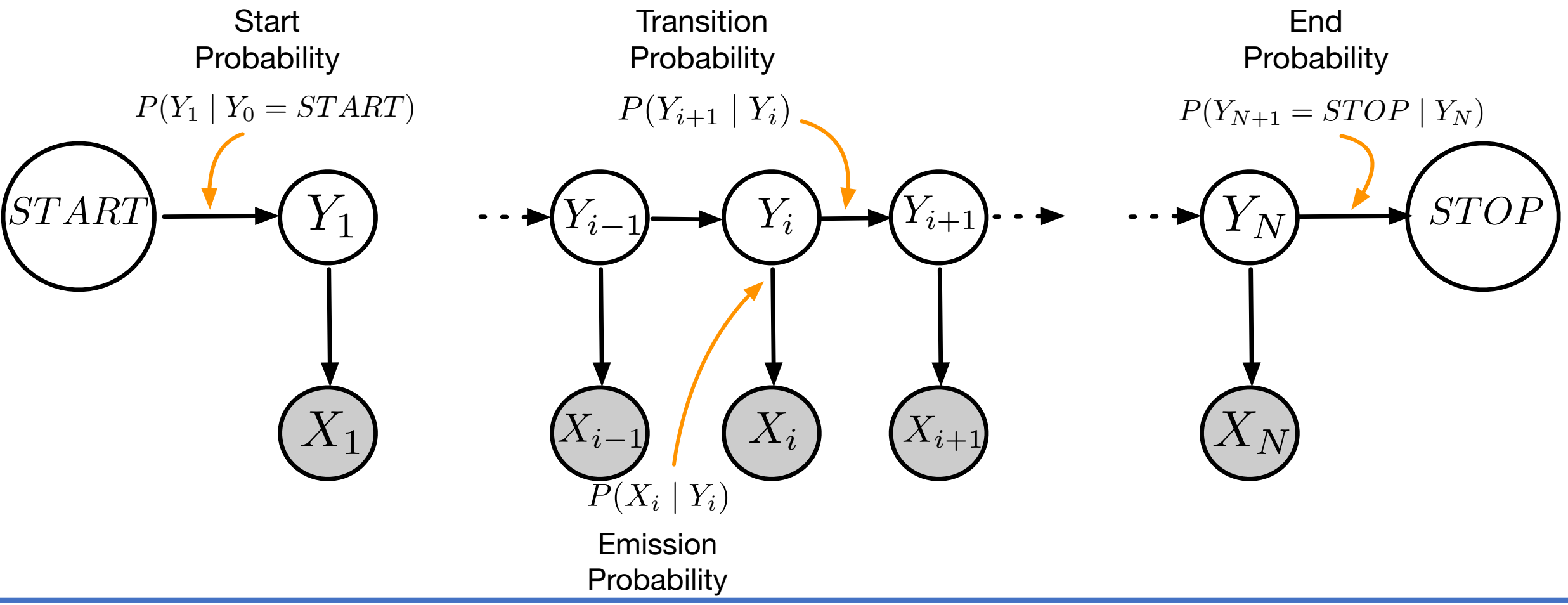
HMM Setting

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 Limitations

HMM Limitations

1. Transition Probabilities and Emission probabilities do not encode any information/knowledge other than information about previous/emitting tag

HMM Limitations

1. Transition Probabilities and Emission probabilities do not encode any information/knowledge other than information about previous/emitting tag

2. Moreover, it is difficult to fit some knowledge into these probabilities.

3. It is difficult to condition prediction of a tag on (lexical/semantic) features which might be useful for prediction.

HMM Limitations

1. Transition Probabilities and Emission probabilities do not encode any information/knowledge other than information about previous/emitting tag

2. Moreover, it is difficult to fit some knowledge into these probabilities.

3. It is difficult to condition prediction of a tag on (lexical/semantic) features which might be useful for prediction.

$$P(Y_i = \text{NOUN} \mid Y_{i-1} = \text{VERB})$$ ✓

HMM Limitations

1. Transition Probabilities and Emission probabilities do not encode any information/knowledge other than information about previous/emitting tag

2. Moreover, it is difficult to fit some knowledge into these probabilities.

3. It is difficult to condition prediction of a tag on (lexical/semantic) features which might be useful for prediction.

$$P(Y_i = \text{NOUN} \mid Y_{i-1} = \text{VERB})$$ ✓

$$P(Y_i = \text{NOUN} \mid Y_{i-1} = \text{VERB},\ \text{CAPITAL}(X_i) = \text{True})$$ ✗

Maximum Entropy Markov Model (MEMM)

- Allow highly flexible representations, allowing features to be easily integrated into the model.

Maximum Entropy Markov Model (MEMM)

- Allow highly flexible representations, allowing features to be easily integrated into the model.
- Also called as ***Log-Linear Tagging Model***
- Discriminative Model

Maximum Entropy Markov Model (MEMM)

- Allow highly flexible representations, allowing features to be easily integrated into the model.
- Also called as ***Log-Linear Tagging Model***
- Discriminative Model

HMM

$$\hat{Y}_{1:N} = \underset{y_{1:N} \in \Lambda}{\operatorname{argmax}} P(X_{1:N}, Y_{1:N} = y_{1:N})$$

$$= \underset{y_{1:N} \in \Lambda}{\operatorname{argmax}} \prod_{i=1}^{N+1} P(Y_i \mid Y_{i-1}) \times P(X_i \mid Y_i)$$

Maximum Entropy Markov Model (MEMM)

- Allow highly flexible representations, allowing features to be easily integrated into the model.
- Also called as ***Log-Linear Tagging Model***
- Discriminative Model

HMM

$$\hat{Y}_{1:N} = \underset{y_{1:N} \in \Lambda}{\operatorname{argmax}} P(X_{1:N}, Y_{1:N} = y_{1:N})$$

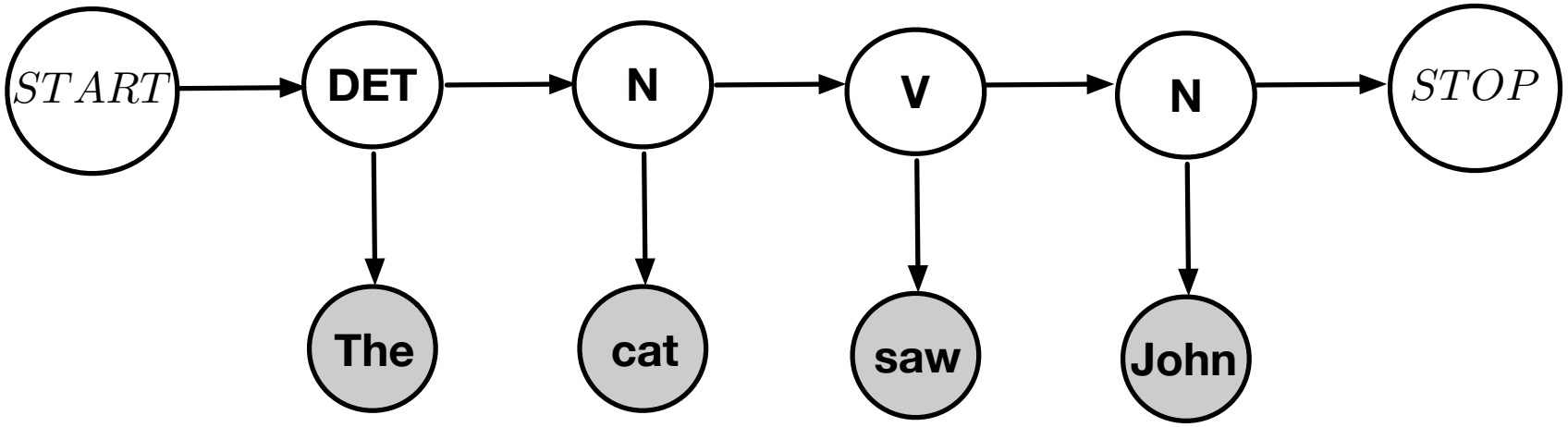
$$= \underset{y_{1:N} \in \Lambda}{\operatorname{argmax}} \prod_{i=1}^{N+1} P(Y_i \mid Y_{i-1}) \times P(X_i \mid Y_i)$$

MEMM

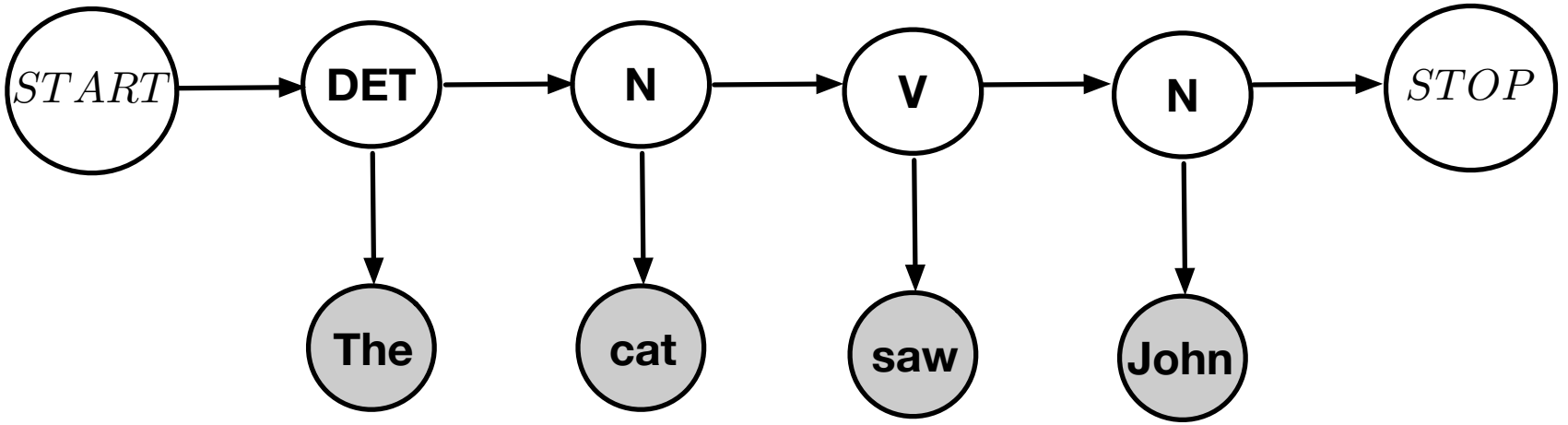
$$\hat{Y}_{1:N} = \underset{y_{1:N} \in \Lambda}{\operatorname{argmax}} P(Y_{1:N} = y_{1:N} \mid X_{1:N})$$

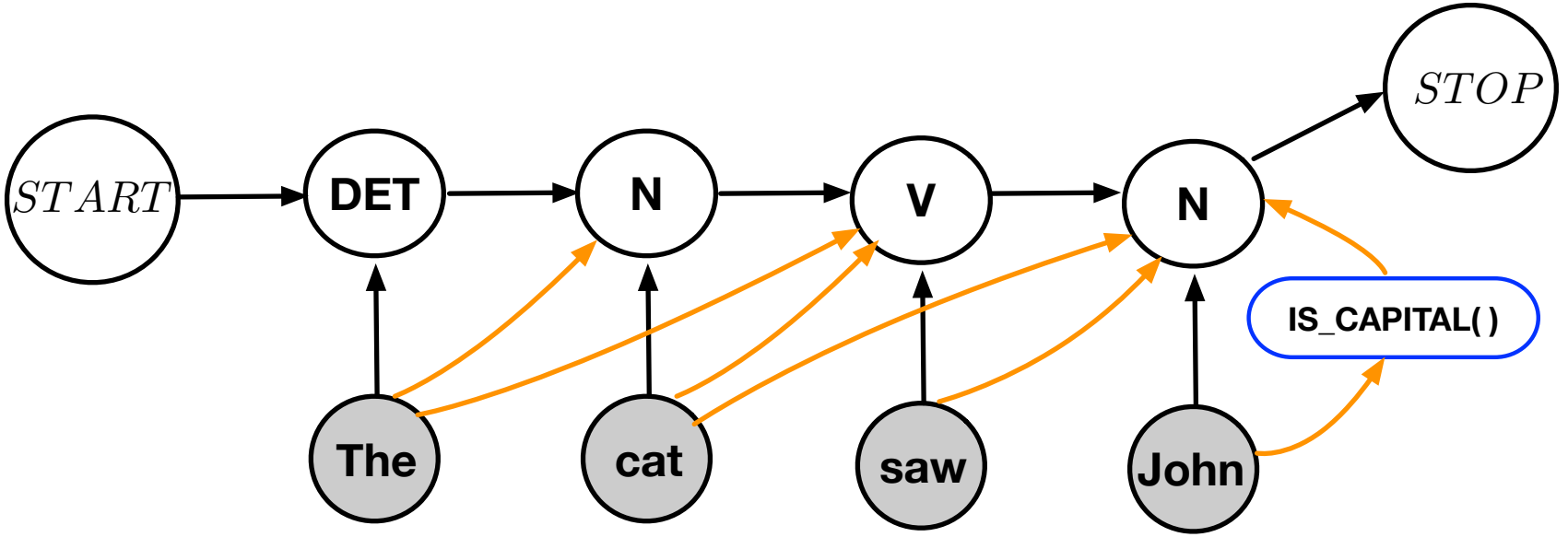
$$= \underset{y_{1:N} \in \Lambda}{\operatorname{argmax}} \prod_{i=1}^{N+1} P(Y_i \mid Y_{i-1}, X_{1:N})$$

HMM vs MEMM



HMM vs MEMM


START
DET
N
V
N
STOP
The
cat
saw
John


START
DET
N
V
N
STOP
IS_CAPITAL( )
The
cat
saw
John

Why Discriminative Models?

- Generative Model:
  - Models joint distribution *P(X, Y)*
  - It models all the dependencies between input and output

Why Discriminative Models?

- Generative Model:
  - Models joint distribution *P(X, Y)*
  - It models all the dependencies between input and output
  - *x* can have high dimensionality and dependencies between variables can be highly complex

Why Discriminative Models?

- Generative Model:
  - Models joint distribution *P(X, Y)*
  - It models all the dependencies between input and output
  - *x* can have high dimensionality and dependencies between variables can be highly complex
  - Modeling dependencies among variables can get intractable

Why Discriminative Models?

- Generative Model:
  - Models joint distribution *P(X, Y)*
  - It models all the dependencies between input and output
  - *x* can have high dimensionality and dependencies between variables can be highly complex
  - Modeling dependencies among variables can get intractable
  - Reducing dependencies between variables can reduce the power of the model

Why Discriminative Models?

- Generative Model:
  - Models joint distribution *P(X, Y)*
  - It models all the dependencies between input and output
  - *x* can have high dimensionality and dependencies between variables can be highly complex
  - Modeling dependencies among variables can get intractable
  - Reducing dependencies between variables can reduce the power of the model
  - If we are interested only in output (hidden states), does it make sense to model joint distribution?

Why Discriminative Models?

- Generative Model:
  - Models joint distribution *P(X, Y)*
  - It models all the dependencies between input and output
  - *x* can have high dimensionality and dependencies between variables can be highly complex
  - Modeling dependencies among variables can get intractable
  - Reducing dependencies between variables can reduce the power of the model
  - If we are interested only in output (hidden states), does it make sense to model joint distribution?
  - Model the conditional distribution directly *P(Y | X)*

Maximum Entropy Markov Model (MEMM)

$$P(Y_1 = y_1, \cdots, Y_N = y_N \mid X_1 = x_1, \cdots, X_N = x_N)$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_1 = y_1, \cdots, Y_N = y_N \mid X_1 = x_1, \cdots, X_N = x_N)$$

$$= \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, \cdots, Y_1 = y_1, X_1 = x_1, \cdots, X_N = x_N)$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_1 = y_1, \cdots, Y_N = y_N \mid X_1 = x_1, \cdots, X_N = x_N)$$

$$= \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, \cdots, Y_1 = y_1, X_1 = x_1, \cdots, X_N = x_N)$$

$$= \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_1 = x_1, \cdots, X_N = x_N)$$

$$= \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$h_i =< y_{i-1}, x_1, \ldots, x_N, i >$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$h_i = < y_{i-1}, x_1 \ldots x_N, i >$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

Three Questions:

1. ***Feature Vectors***: How do we define feature vector?
2. ***Learning Problem***: How do we learn parameters?
3. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

Three Questions:

1. ***Feature Vectors***: How do we define feature vector?
2. ***Learning Problem***: How do we learn parameters?
3. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d$$

$$h_i = < y_{i-1}, x_1 \ldots x_N, i >$$

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d$$

$$h_i = < y_{i-1}, x_1 \ldots x_N, i >$$

- Each feature can capture any information in history *h* in conjunction with tag *y*

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d$$

$$h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

- Each feature can capture any information in history *h* in conjunction with tag *y*
- A much richer set of features can be employed for the tagging task

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

WORD/TAG FEATURES

$$f_{100}(h, y) = \begin{cases} 1 & \text{If } x_i = \texttt{base} \text{ and } y = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

Ratnaparkhi, 96

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

SUFFIX FEATURES

$$f_{101}(h, y) = \begin{cases} 1 & \text{If } x_i \text{ ends in } \texttt{ing} \text{ and } y = \text{VBG} \\ 0 & \text{otherwise} \end{cases}$$

Ratnaparkhi, 96

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

PREFIX FEATURES

$$f_{102}(h, y) = \begin{cases} 1 & \text{If } x_i \text{ starts with } \texttt{pre} \text{ and } y = \text{NN} \\ 0 & \text{otherwise} \end{cases}$$

Ratnaparkhi, 96

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

BI-GRAM FEATURES

$$f_{103}(h, y) = \begin{cases} 1 & \text{If } < y_{i-1}, y_i > = < \text{DT,NN} > \\ 0 & \text{otherwise} \end{cases}$$

Ratnaparkhi, 96

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

UNI-GRAM FEATURES

$$f_{104}(h, y) = \begin{cases} 1 & \text{If } < y > = < \text{NN} > \\ 0 & \text{otherwise} \end{cases}$$

Ratnaparkhi, 96

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

CONTEXTUAL FEATURES

$$f_{105}(h, y) = \begin{cases} 1 & \text{If previous word } x_{i-1} = \texttt{the} \text{ and } y = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

Ratnaparkhi, 96

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

CONTEXTUAL FEATURES

$$f_{106}(h, y) = \begin{cases} 1 & \text{If next word } x_{i+1} = \texttt{the} \text{ and } y = \text{VB} \\ 0 & \text{otherwise} \end{cases}$$

Ratnaparkhi, 96

MEMM: Feature Vectors

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

$$f(h_i, y_i) \in \mathbb{R}^d \qquad h_i =< y_{i-1}, x_1 \ldots x_N, i >$$

OTHER FEATURES

- Spelling features which consider weather a word being tagged contains number, contains a hyphen, or contains an upper-case letter
- Contextual features that consider the word at x_{i-2} and x_{i+2} in conjunction with the current tag *y*

Ratnaparkhi, 96

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

Three Questions:

1. ***Feature Vectors***: How do we define feature vector?
2. ***Learning Problem***: How do we learn parameters?
3. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?

Parameter Estimation in MEMM

TRAINING DATA $(x^{(k)}, y^{(k)}) \qquad k = 1, \ldots, m$

$$x^{(k)} = \{x_1^{(k)}, \ldots, x_{N_k}^{(k)}\}$$

$$y^{(k)} = \{y_1^{(k)}, \ldots, y_{N_k}^{(k)}\}$$

$$p(y_i^{(k)} \mid h_i^{(k)}; \theta) = \frac{exp(\theta^T f(h_i^{(k)}, y_i^{(k)}))}{\sum_{y' \in K} exp(\theta^T f(h_i^{(k)}, y'))}$$

Parameter Estimation in MEMM

$$p(y_i^{(k)} \mid h_i^{(k)}; \theta) = \frac{exp(\theta^T f(h_i^{(k)}, y_i^{(k)}))}{\sum_{y' \in K} exp(\theta^T f(h_i^{(k)}, y'))}$$

How do we estimate parameters?

Parameter Estimation in MEMM

$$p(y_i^{(k)} \mid h_i^{(k)}; \theta) = \frac{exp(\theta^T f(h_i^{(k)}, y_i^{(k)}))}{\sum_{y' \in K} exp(\theta^T f(h_i^{(k)}, y'))}$$

$$\mathcal{L}(\theta) = \sum_{k=1}^{m} \sum_{i=1}^{N_k} log\ p(y_i^{(k)} \mid h_i^{(k)}; \theta) - \frac{\lambda}{2} \sum_{j=1}^{d} \theta_j^2$$

Parameter Estimation in MEMM

$$p(y_i^{(k)} \mid h_i^{(k)}; \theta) = \frac{exp(\theta^T f(h_i^{(k)}, y_i^{(k)}))}{\sum_{y' \in K} exp(\theta^T f(h_i^{(k)}, y'))}$$

$$\mathcal{L}(\theta) = \sum_{k=1}^{m} \sum_{i=1}^{N_k} log\ p(y_i^{(k)} \mid h_i^{(k)}; \theta) - \frac{\lambda}{2} \sum_{j=1}^{d} \theta_j^2$$

$$\theta^* = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmax}}\, \mathcal{L}(\theta)$$

Maximum Entropy Markov Model (MEMM)

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

Three Questions:

1. ***Feature Vectors***: How do we define feature vector?
2. ***Learning Problem***: How do we learn parameters?
3. ***Decoding Problem***: Given the observed text what is the hidden POS sequence that best explains the observation?

DECODING IN MEMM

$$P(Y_{1:N} = y_{1:N} \mid X_{1:N} = x_{1:N}) = \prod_{i=1}^{N} P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N})$$

$$P(Y_i = y_i \mid Y_{i-1} = y_{i-1}, X_{1:N}) = \frac{exp(\theta^T f(h_i, y_i))}{\sum_{y' \in K} exp(\theta^T f(h_i, y'))}$$

DECODING

$$\underset{y_1,\ldots,y_N}{\text{argmax}} \; P(Y_1 = y_1, \cdots, Y_N = y_N \mid X_1 = x_1, \cdots, X_N = x_N)$$

$$\underset{y_1,\ldots,y_N}{\text{argmax}} \prod_{i=1}^{N} p(y_i \mid h_i; \theta)$$

DECODING IN MEMM

DECODING IN MEMM

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid h_i; \theta)$$

DECODING IN HMM

DECODING IN MEMM

DECODING IN MEMM

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid h_i; \theta)$$

DECODING IN HMM

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid y_{i-1}) \times p(x_i \mid y_i)$$

DECODING IN MEMM: VITERBI

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid h_i; \theta)$$

$$\text{viterbi}(k, c_k) = \max_{y_1,\ldots,y_k} \prod_{i=1}^{k} p(y_i \mid h_i; \theta)$$

DECODING IN MEMM: VITERBI

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid h_i; \theta)$$

$$\text{viterbi}(k, c_k) = \max_{y_1,\ldots,y_k} \prod_{i=1}^{k} p(y_i \mid h_i; \theta)$$

$$\text{viterbi}(k, c_k) = \max_{c_l} \text{viterbi}(k-1, c_l) \times p(Y_k = c_k \mid h_k; \theta)$$

DECODING IN MEMM: VITERBI

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid h_i; \theta)$$

$$\text{viterbi}(k, c_k) = \max_{y_1,\ldots,y_k} \prod_{i=1}^{k} p(y_i \mid h_i; \theta)$$

$$\text{viterbi}(k, c_k) = \max_{c_l} \text{viterbi}(k-1, c_l) \times p(Y_k = c_k \mid h_k; \theta)$$

HMM:

DECODING IN MEMM: VITERBI

$$\underset{y_1,\ldots,y_N}{\operatorname{argmax}} \prod_{i=1}^{N} p(y_i \mid h_i; \theta)$$

$$\text{viterbi}(k, c_k) = \max_{y_1,\ldots,y_k} \prod_{i=1}^{k} p(y_i \mid h_i; \theta)$$

$$\text{viterbi}(k, c_k) = \max_{c_l} \text{viterbi}(k-1, c_l) \times p(Y_k = c_k \mid h_k; \theta)$$

HMM:

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

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$\quad For\ l = 1, \ldots, K$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$\quad For\ l = 1, \ldots, K$

$$\text{viterbi}(k, c_l) = \max_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$For\ l = 1, \ldots, K$

$$\text{viterbi}(k, c_l) = \max_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$$\text{bp}(k, c_l) = \operatorname*{argmax}_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$\quad For\ l = 1, \ldots, K$

$$\text{viterbi}(k, c_l) = \max_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$$\text{bp}(k, c_l) = \operatorname*{argmax}_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$\quad end\ for$

$end\ for$

Set: $y_N = \operatorname*{argmax}_{c_j} \text{viterbi}(N-1, c_j)$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$For\ l = 1, \ldots, K$

$$\text{viterbi}(k, c_l) = \max_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$$\text{bp}(k, c_l) = \operatorname*{argmax}_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$end\ for$

$end\ for$

Set: $y_N = \operatorname*{argmax}_{c_j} \text{viterbi}(N-1, c_j)$

$For\ k = (N-1), \ldots, 1$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$\quad For\ l = 1, \ldots, K$

$$\text{viterbi}(k, c_l) = \max_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$$\text{bp}(k, c_l) = \operatorname*{argmax}_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$\quad end\ for$

$end\ for$

Set: $y_N = \operatorname*{argmax}_{c_j} \text{viterbi}(N-1, c_j)$

$For\ k = (N-1), \ldots, 1$

$\quad y_k = bp(k+1, y_{k+1})$

$end\ for$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$\quad For\ l = 1, \ldots, K$

$$\text{viterbi}(k, c_l) = \max_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$$\text{bp}(k, c_l) = \operatorname*{argmax}_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$\quad end\ for$

$end\ for$

Set: $y_N = \operatorname*{argmax}_{c_j} \text{viterbi}(N-1, c_j)$

$For\ k = (N-1), \ldots, 1$

$\quad y_k = bp(k+1, y_{k+1})$

$end\ for$

Return $y_1, \ldots, y_N$

VITERBI ALGORITHM

Initialization: $\text{viterbi}(0, START) = 1$

$For\ k = 1, \ldots, N$

$\quad For\ l = 1, \ldots, K$

$$\text{viterbi}(k, c_l) = \max_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$$\text{bp}(k, c_l) = \operatorname*{argmax}_{c_j} \left(\text{viterbi}(k-1, c_j) \times p(Y_k = c_l \mid h_k; \theta)\right)$$

$\quad end\ for$

$end\ for$

Set: $y_N = \operatorname*{argmax}_{c_j} \text{viterbi}(N-1, c_j)$

$For\ k = (N-1), \ldots, 1$

$\quad y_k = bp(k+1, y_{k+1})$

$end\ for$

Return $y_1, \ldots, y_N$

Summary

1. It is difficult to consider features in HMM models.
2. MEMMs make it possible to include rich set of features in sequence prediction
3. MEMMs are discriminative models
4. Features can be included in MEMMs model via indicator functions
5. Parameters can be estimated using MLE
6. Decoding in MEMMs can be done via Viterbi Algorithm.

References

1. Michael Collin's NLP Lecture Notes: http://www.cs.columbia.edu/~mcollins/fall2014-loglineartaggers.pdf
2. Chapter 6, Speech and Language Processing, Dan Jurafsky and James Martin
3. A Maximum Entropy Model for Part of Speech Tagging: https://www.aclweb.org/anthology/W96-0213.pdf