TTIC 31190: Natural Language Processing

Lecture 8: Sequence Labeling

Fall 2023

Announcement

- Message from our grader Kangrui:
  - It's preferable to use Jupyter notebooks for assignments and submit the exported .pdf report
  - It's fine to note your late-day usage in your report

- Freda will be out of town tomorrow
  - Joe's office hours: Tue 1:30-2:30 pm TTIC 4th floor open space (unchanged)
  - Freda's office hours this week: Thu 1:30-2:30 pm TTIC 4th floor open space

Recap: Transformers

- Idea: every token attends every other token in a sequence, and transform (noncontextualized) word token embeddings into contextualized word embeddings

$$\mathbf{E} = (emb(w_1), \ldots, emb(w_k)) \in \mathbb{R}^{d_1 \times k}$$

$$\mathbf{K} = \mathbf{W}_k \mathbf{E} \quad \mathbf{W}_k \in \mathbb{R}^{d_2 \times d_1}, \mathbf{K} \in \mathbb{R}^{d_2 \times k}$$

$$\mathbf{Q} = \mathbf{W}_q \mathbf{E} \quad \mathbf{W}_q \in \mathbb{R}^{d_2 \times d_1}, \mathbf{Q} \in \mathbb{R}^{d_2 \times k}$$

$$\mathbf{V} = \mathbf{W}_v \mathbf{E} \quad \mathbf{W}_v \in \mathbb{R}^{d_3 \times d_1}, \mathbf{V} \in \mathbb{R}^{d_3 \times k}$$

$$\tilde{\mathbf{E}} = \mathbf{V}\text{softmax}\left(\frac{\mathbf{K}^\top \mathbf{Q}}{\sqrt{d_2}}\right) \in \mathbb{R}^{d_3 \times k}$$

Contextualized Word Embeddings

- Each word has a fixed-dimensional vector
- The vector has information about other words in the sentence

$$\mathbf{E} = (emb(w_1), \ldots, emb(w_k)) \in \mathbb{R}^{d_1 \times k}$$

$$\mathbf{K} = \mathbf{W}_k \mathbf{E} \quad \mathbf{W}_k \in \mathbb{R}^{d_2 \times d_1}, \mathbf{K} \in \mathbb{R}^{d_2 \times k}$$

$$\mathbf{Q} = \mathbf{W}_q \mathbf{E} \quad \mathbf{W}_q \in \mathbb{R}^{d_2 \times d_1}, \mathbf{Q} \in \mathbb{R}^{d_2 \times k}$$

$$\mathbf{V} = \mathbf{W}_v \mathbf{E} \quad \mathbf{W}_v \in \mathbb{R}^{d_3 \times d_1}, \mathbf{V} \in \mathbb{R}^{d_3 \times k}$$

$$\tilde{\mathbf{E}} = \mathbf{V}\mathrm{softmax}\left(\frac{\mathbf{K}^\top \mathbf{Q}}{\sqrt{d_2}}\right) \in \mathbb{R}^{d_3 \times k}$$

Recap: POS Tagging with Contextualized Word Embeddings

- Break down the problem to k independent classification problems. (k: number of words in the sentence)
- Classify contextualized word embeddings with an NN (e.g., an MLP)

$$\mathrm{POS}(w_i) = \arg\max_y \mathrm{score}(\mathbf{h}_i, y; \mathbf{w})$$

contextualized word embedding of w_i

POS tag

This Lecture

- Hidden Markov Models
  - **Formulation and properties**
  - Learning: estimate the parameters
  - Inference: finding the highest-scoring sequence of hidden variable values
- Conditional Random Fields

Sequence Labeling as Structured Prediction

Inference: solve arg max

Modeling: define score function

$$\mathrm{POS}(\mathbf{x}; \mathbf{w}) = \arg\max_{\boldsymbol{y}} \mathrm{score}(\mathbf{x}, \boldsymbol{y}; \mathbf{w})$$

Learning: choose parameter

- $\boldsymbol{y}$: a sequence of POS tags
- Unlike classification, inference is no longer trivial ($|Y|^k$ possibilities!)

Sequence Labeling as Structured Prediction

Modeling: define score function

$$\mathrm{POS}(\mathbf{x}; \mathbf{w}) = \arg\max_{\boldsymbol{y}} \mathrm{score}(\mathbf{x}, \boldsymbol{y}; \mathbf{w})$$

Hidden Markov Models

- Used in (conventional) NLP, speech processing, computational biology, and many other areas
- Good starting point for learning graphical models

- TTIC 31180: Probabilistic Graphical Models is offered in Spring 2024

Recap: Independence

- Two random variables X and Y are independent if

$$P(X = x, Y = y) = P(X = x)P(Y = y)$$

  for all values x and y

- We write this as $X \perp Y$

Recap: Conditional Independence

- Two random variables X and Y are conditionally independent given a third variable Z if

$$P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z)$$

for all values x, y and z

- We write this as $X \perp Y \mid Z$

- Example: Height and vocabulary are independent conditioned on age.

Recap: Conditional Independence

- Two random variables X and Y are conditionally independent given a third variable Z if

$$P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z)$$

for all values x, y and z

$$P(x, y, z) = P(z)P(x, y \mid z)$$
$$= P(z)P(x \mid z)\boxed{P(y \mid z)}$$

should be $P(y \mid x, z)$ w/o conditional independence

Markov Chain

- Stochastic model: a sequence of possible events
  - Probability of each event depends only on the state attained in the previous event

$$P(X_t \mid X_1, X_2, \ldots, X_{t-1}) = P(X_t \mid X_{t-1})$$

$$X_t \perp X_{t-2}, \ldots, X_1 \mid X_{t-1}$$

Andrey Markov

- Example: board games played with dice
  - At each move, a player rolls the dice and move their piece some steps forward
  - X_t = position of the piece after t rolls

Markov Assumption

$$P(w_t \mid w_1, \ldots, w_{t-1}) = P(w_t \mid w_{t-1})$$

$$P(table \mid a\ cat\ is\ sitting\ on\ the) = P(table \mid the)$$

This is obviously imperfect: we should consider longer dependencies

But we can model sentence probability with fewer parameters

$$P(w_1, \ldots, w_k) = P(w_1) \prod_{t=2}^{k} P(w_t \mid w_1, \ldots, w_{t-1})$$

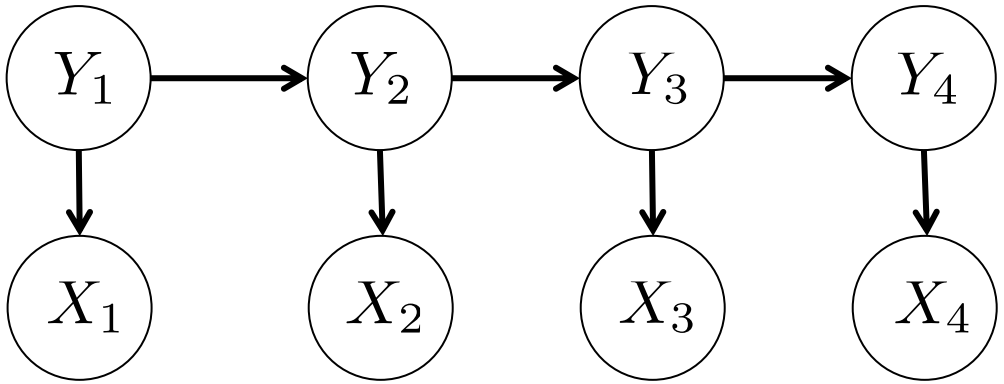
$$= P(w_1) \prod_{t=2}^{k} P(w_t \mid w_{t-1})$$

Hidden Markov Models

- Modeling joint probability of the observable sequence $X_1, \ldots, X_k$ and hidden variables $Y_1, \ldots, Y_k$ with the following assumptions

$$X_t \perp X_1, \ldots, X_{t-1}, Y_1, \ldots, Y_{t-1} \mid Y_t$$

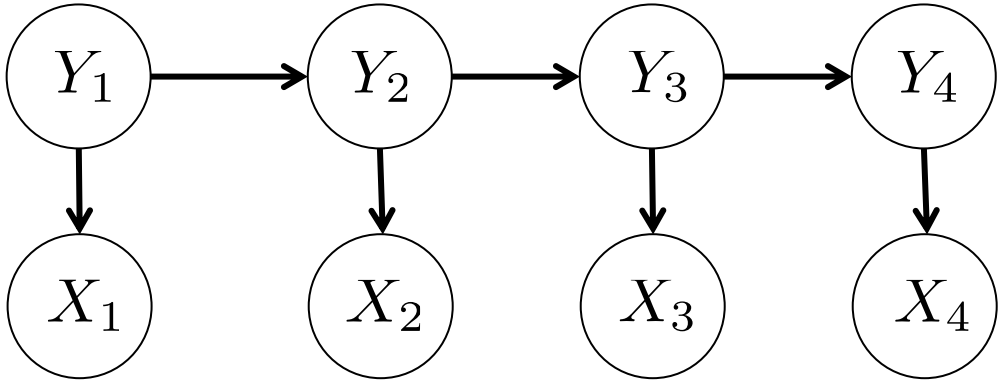
$$Y_t \perp X_1, \ldots, X_{t-1}, Y_1, \ldots, Y_{t-2} \mid Y_{t-1}$$



- An instantiation of Bayesian network: graphical model representing conditional dependency with a directed acyclic graph (DAG)

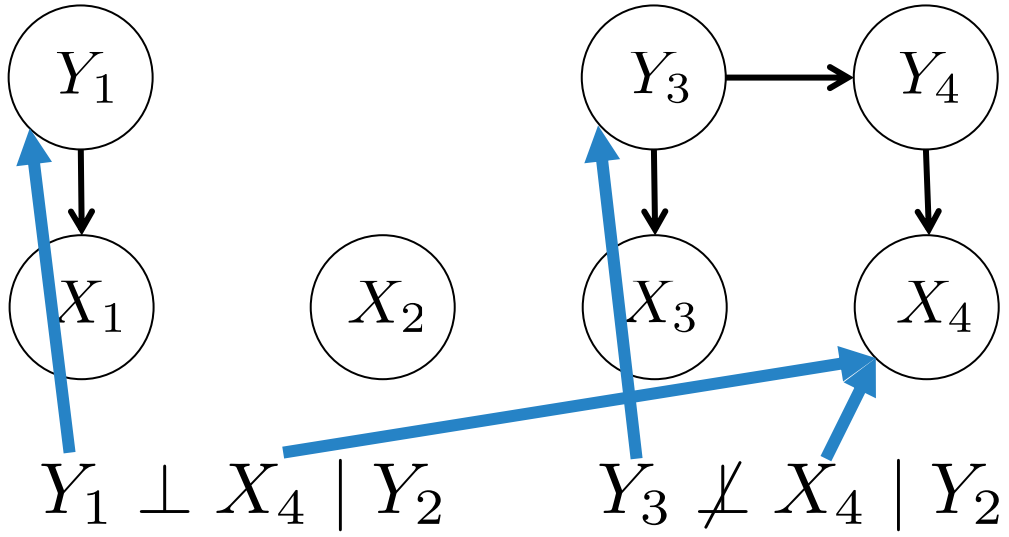
Bayesian Networks

- Distances are not meaningful
- $A \perp B \mid C \Leftrightarrow A$ and B are disconnected after removing nodes corresponding to variables in C and all connected edges



Bayesian Networks

- Distances are not meaningful
- $A \perp B \mid C \Leftrightarrow A$ and B are disconnected after removing nodes corresponding to variables in C and all connected edges

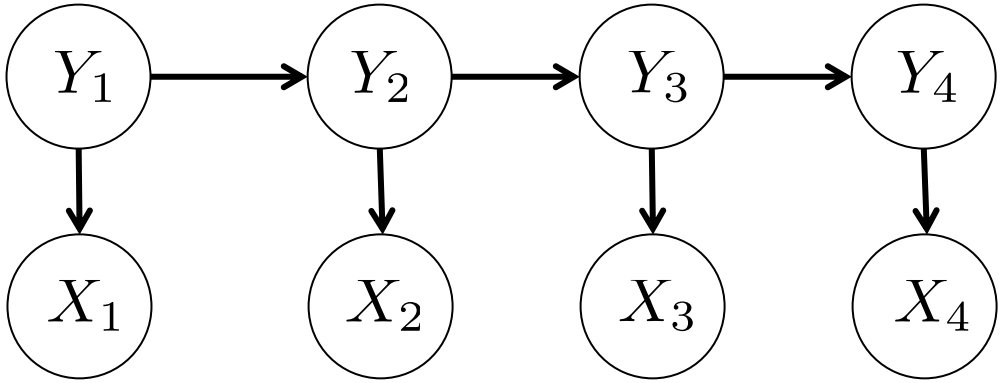


Hidden Markov Models

$$P(X_1, \ldots, X_k, Y_1, \ldots, Y_k)$$

$$= P(Y_1)P(X_1 \mid Y_1) \prod_{t=2}^{k} P(Y_t \mid X_{1:t-1}, Y_{1:t-1}) P(X_t \mid X_{1:t-1}, Y_{1:t})$$

$$= P(Y_1)P(X_1 \mid Y_1) \prod_{t=2}^{k} P(Y_t \mid Y_{t-1}) P(X_t \mid Y_t)$$



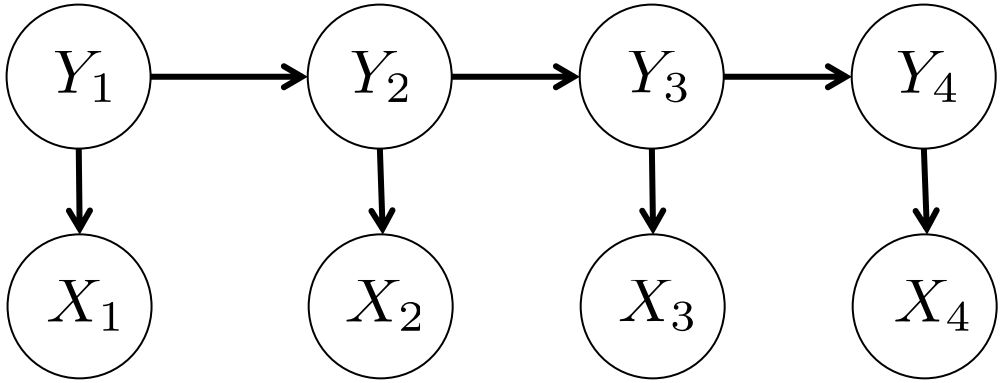
Hidden Markov Models for POS Tagging

- Y_t: part-of-speech tags, X_t: words

$$P(X_1, \ldots, X_k, Y_1, \ldots, Y_k) = P(Y_1)P(X_1 \mid Y_1) \prod_{t=2}^{k} P(Y_t \mid Y_{t-1}) P(X_t \mid Y_t)$$

transition probability

emission probability



The transition and emission probabilities can be estimated by counting.

This Lecture

- Hidden Markov Models
  - Formulation and properties
  - **Learning: estimate the parameters**
  - Inference: finding the highest-scoring sequence of hidden variable values
- Conditional Random Fields

Sequence Labeling as Structured Prediction

$$\mathrm{POS}(\mathbf{x}; \mathbf{w}) = \arg\max_{\boldsymbol{y}} \mathrm{score}(\mathbf{x}, \boldsymbol{y}; \mathbf{w})$$

Learning: choose parameter

Hidden Markov Models: Parameter Estimation

- We will denote the transition probability by P_T and emission probability by P_E

$$P_T(y \mid y') = \frac{\text{count}(y', y)}{\text{count}(y')} \qquad P_E(x \mid y) = \frac{\text{count}(x, y)}{\text{count}(y)}$$

$$P_T(\text{verb} \mid \text{noun}) = \frac{\text{count}(\text{noun}, \text{verb})}{\text{count}(\text{noun})}$$

$$P_E(\text{cat} \mid \text{noun}) = \frac{\text{count}(\text{cat}, \text{noun})}{\text{count}(\text{noun})}$$

Stopping Probability

- Y_t: part-of-speech tags, X_t: words
  Let $P_T(\, Y_1 \mid Y_0 \,) = P(Y_1)$

$$p(\boldsymbol{x}, \boldsymbol{y}) = \prod_{t=1}^{k} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$\downarrow$$

$$p(\boldsymbol{x}, \boldsymbol{y}) = \prod_{t=1}^{k} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t) P_T(\boxed{\langle \text{eos} \rangle} \mid y_k)$$

End of sentence symbol

- Purpose: have all sentences' probability sum up to 1.

Stopping Probability

$$\sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} \prod_{t=1}^{k} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \sum_{\boldsymbol{x}_{1:k-1}} \sum_{\boldsymbol{y}_{1:k}} \sum_{x_k} P_E(x_k \mid y_k) P_T(y_k \mid y_{k-1}) \prod_{t=1}^{k-1} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

Stopping Probability

$$\sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} \prod_{t=1}^{k} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \sum_{\boldsymbol{x}_{1:k-1}} \sum_{\boldsymbol{y}_{1:k}} \boxed{\sum_{x_k} P_E(x_k \mid y_k)} P_T(y_k \mid y_{k-1}) \prod_{t=1}^{k-1} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

Apply distributivity of multiplication over addition

Stopping Probability

$$\sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} \prod_{t=1}^{k} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \sum_{\boldsymbol{x}_{1:k-1}} \sum_{\boldsymbol{y}_{1:k}} \sum_{x_k} P_E(x_k \mid y_k) P_T(y_k \mid y_{k-1}) \prod_{t=1}^{k-1} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \sum_{\boldsymbol{x}_{1:k-1}} \sum_{\boldsymbol{y}_{1:k}} P_T(y_k \mid y_{k-1}) \prod_{t=1}^{k-1} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

Stopping Probability

$$\sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} \prod_{t=1}^{k} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \sum_{\boldsymbol{x}_{1:k-1}} \sum_{\boldsymbol{y}_{1:k}} \sum_{x_k} P_E(x_k \mid y_k) P_T(y_k \mid y_{k-1}) \prod_{t=1}^{k-1} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \sum_{\boldsymbol{x}_{1:k-1}} \sum_{\boldsymbol{y}_{1:k}} P_T(y_k \mid y_{k-1}) \prod_{t=1}^{k-1} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \sum_{\boldsymbol{x}_{1:k-1}} \sum_{\boldsymbol{y}_{1:k-1}} \sum_{y_k} P_T(y_k \mid y_{k-1}) \prod_{t=1}^{k-1} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)$$

$$= \ldots = 1$$

Stopping Probability

$$\sum_{\boldsymbol{x}_{1:k}} \sum_{\boldsymbol{y}_{1:k}} p(\boldsymbol{x}, \boldsymbol{y}) = 1$$

- This means sequences with any fixed length k have a total probability to 1.

$$p(\boldsymbol{x}, \boldsymbol{y}) = \prod_{t=1}^{k} P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t) P_T(\langle \text{eos} \rangle \mid y_k)$$

$$P_T(\langle \text{eos} \rangle \mid y) = \frac{\text{count}(y, \langle \text{eos} \rangle)}{\text{count}(y)}$$

This Lecture

- Hidden Markov Models
  - Formulation and properties
  - Learning: estimate the parameters
  - **Inference: finding the highest-scoring sequence of hidden variable values**

- Conditional Random Fields

Sequence Labeling as Structured Prediction

Inference: solve arg max

$$\mathrm{POS}(\mathbf{x}; \mathbf{w}) = \arg\max_{\boldsymbol{y}} \mathrm{score}(\mathbf{x}, \boldsymbol{y}; \mathbf{w})$$

- Given a sentence, what is the highest-scoring POS tag sequence?
- $\boldsymbol{y}$: a sequence of POS tags
- Unlike classification, inference is no longer trivial ($|Y|^k$ possibilities!)

Hidden Markov Model: Inference

- Find the highest-scoring $\boldsymbol{y}$

$$\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w}) \leftarrow \text{joint probability}$$

$$= \prod_{t=1}^{k} \left(P_T(y_t \mid y_{t-1}) P_E(x_t \mid y_t)\right) P_T(\langle \text{eos} \rangle \mid y_k)$$

$$= \log P_T(\langle \text{eos} \rangle \mid y_k) + \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

Computation difficulty comes from this part

Hidden Markov Model: Inference

- Dynamic programming: recursively break down the problem into subproblems with the same formulation

$$\underset{y_1,\ldots,y_k}{\arg\max}\ \log P_T(\langle \text{eos} \rangle \mid y_k) + \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

Can be done by enumerating values of Y_k

Part to be broken down

Hidden Markov Model: Inference

- Dynamic programming: recursively break down the problem into subproblems with the same formulation

$$\sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

$$= P_T(y_k \mid y_{k-1}) + \log P_E(x_k \mid y_k)$$

$$+ \sum_{t=1}^{k-1} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

Hidden Markov Model: Inference

- Dynamic programming: recursively break down the problem into subproblems with the same formulation

$$\max_{y_{1:k}} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

$$\rightarrow \max_{y_k} \Bigg(P_T(y_k \mid y_{k-1}) + \log P_E(x_k \mid y_k) + \max_{y_{1:k-1}} \sum_{t=1}^{k-1} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t) \Bigg)$$

Hidden Markov Model: Inference

- Dynamic programming: recursively break down the problem into subproblems with the same formulation

$$\max_{y_{1:k}} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

$$\rightarrow \max_{y_k} \Bigg(P_T(y_k \mid y_{k-1}) + \log P_E(x_k \mid y_k) + \max_{y_{1:k-1}} \sum_{t=1}^{k-1} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t) \Bigg)$$

Not a well-formed equation

Hidden Markov Model: Inference

- Dynamic programming: define state of subproblem

$f_{k,y}$: maximum value of $\sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$ with $Y_k = y$

$$\max_{y_{1:k}} \log P_T(\langle \text{eos} \rangle \mid y_k) + \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

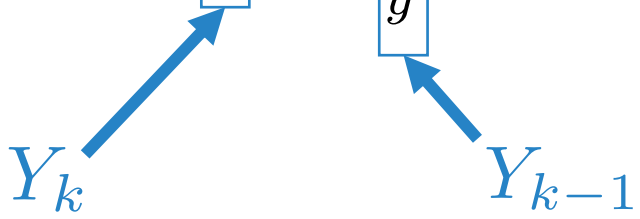
$$= \max_{y} \log P_T(\langle \text{eos} \rangle \mid Y_k = y) + f_{k,y}$$

Hidden Markov Model: Inference

- Dynamic programming: define state of subproblem

$f_{k,y}$: maximum value of $\sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$ with $Y_k = y$

$$f_{k,y} = \max_{y'} \log P_T(y \mid y') + \log P_E(X_k \mid y) + f_{k-1,y'}$$



Hidden Markov Model: Inference

- Recall that our goal is to predict the highest scoring $\boldsymbol{y}$

$f_{k,y}$: maximum value of $\sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$ with $Y_k = y$

$$f_{k,y} = \max_{y'} \log P_T(y \mid y') + \log P_E(x_k \mid y) + f_{k-1,y'}$$

$$g_{k,y} = \arg\max_{y'} \log P_T(y \mid y') + \log P_E(x_k \mid y) + f_{k-1,y'}$$

Which y' gives the optimal $f_{k,y}$

Viterbi Algorithm

- Input: observation (sequence of words) $\boldsymbol{x} = x_1, \ldots, x_k$
- Output: the highest-scoring POS tags $\arg\max_{\boldsymbol{y}} P(\boldsymbol{y} \mid \boldsymbol{x})$

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

$f_{t,y} \leftarrow -\infty$ for all y
for t in [1..k]:
  for y in Y:
    for y' in Y:
      $f_{t,y} = \max\left(f_{t,y}, \log P_T(y \mid y') + \log P_E(X_t \mid y) + f_{t-1,y'}\right)$
      if $f_{t,y}$ is updated in the above line: $g_{t,y} = y'$

Time complexity: $\mathcal{O}(k|Y|^2)$
Memory complexity: $\mathcal{O}(k|Y|)$

Viterbi Algorithm: Example

Emission probability

$y\backslash x$	a	b
0	0.6	0.4
1	0.2	0.8

Transition probability

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	0.9

$P(y_1 \mid y_0)$

start	0	1
	0.5	0.5

$P(\langle eos \rangle \mid y)$

$\langle eos \rangle$	0	1
	0.5	0.5

- Observation $\boldsymbol{x} = a, b$
- What is the most probable sequence of hidden variables $\boldsymbol{y}$?

Viterbi Algorithm: Example

$y\backslash x$	a	b
0	0.6	0.4
1	0.2	0.8

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

Viterbi Algorithm: Example

$y\backslash x$	a	b
0	**0.6**	0.4
1	0.2	0.8

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

$f_{1,0}$

$$f_{1,0} = \log P(Y_1 = 0) + \log P_E(X_1 = a \mid Y_1 = 0)$$
$$= -1.204$$

Viterbi Algorithm: Example

$y\backslash x$	a	b
0	0.6	0.4
1	**0.2**	0.8

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	0.9

start	0	1
	0.5	**0.5**

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$

$f_{1,0}$

-1.204

$f_{1,1}$

$$f_{1,0} = \log P(Y_1 = 1) + \log P_E(X_1 = a \mid Y_1 = 1)$$
$$= -2.303$$

Viterbi Algorithm: Example

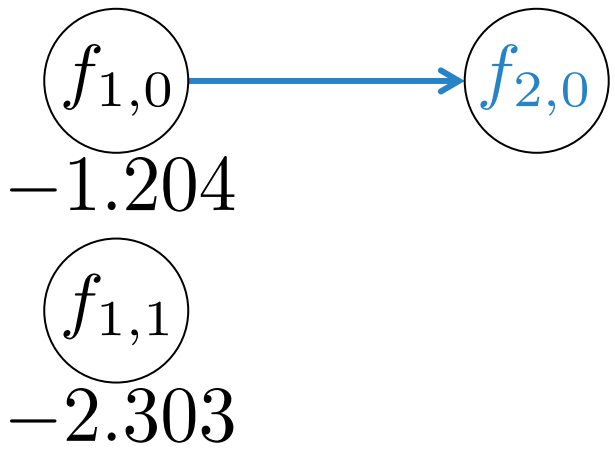
$y\backslash x$	a	b
0	0.6	**0.4**
1	0.2	0.8

$y_{t-1}\backslash y_t$	0	1
0	**0.9**	0.1
1	0.1	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,0} = \max\left(-\infty, \log P_T(0 \mid 0) + \log P_E(b \mid 0) + f_{1,0}\right)$$
$$= -2.226$$

Viterbi Algorithm: Example

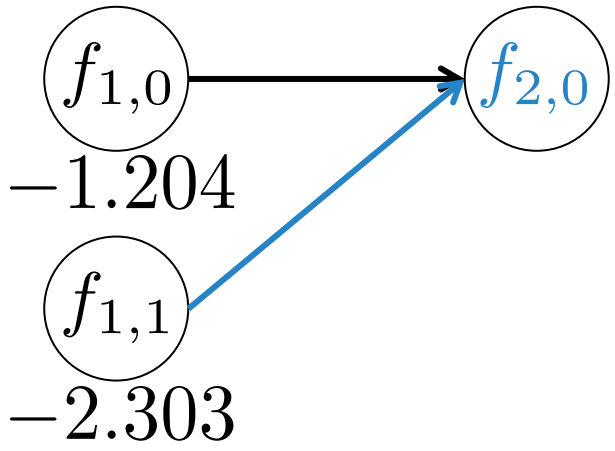
$y\backslash x$	a	b
0	0.6	**0.4**
1	0.2	0.8

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	**0.1**	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,0} = \max(-\infty, \log P_T(0 \mid 0) + \log P_E(b \mid 0) + f_{1,0})$$
$$= -2.226$$
$$f_{2,0} = \max(f_{2,0}, \log P_T(0 \mid 1) + \log P_E(b \mid 0) + f_{1,1})$$
$$= \max(-5.522, -2.226) = -2.226$$

Viterbi Algorithm: Example

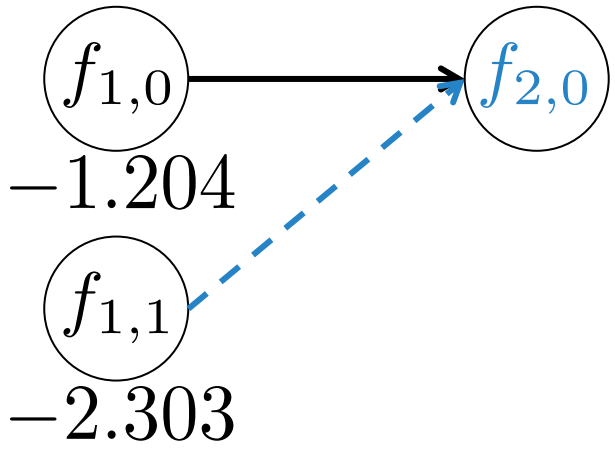
$y\backslash x$	a	b
0	0.6	**0.4**
1	0.2	0.8

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	**0.1**	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,0} = \max(-\infty, \log P_T(0 \mid 0) + \log P_E(b \mid 0) + f_{1,0})$$
$$= -2.226$$
$$f_{2,0} = \max(f_{2,0}, \log P_T(0 \mid 1) + \log P_E(b \mid 0) + f_{1,1})$$
$$= \max(-5.522, -2.226) = -2.226$$

Viterbi Algorithm: Example

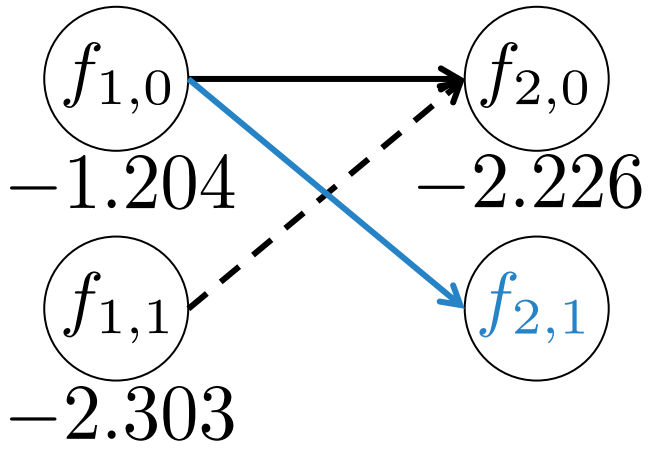
$y\backslash x$	a	b
0	0.6	0.4
1	0.2	**0.8**

$y_{t-1}\backslash y_t$	0	1
0	0.9	**0.1**
1	0.1	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,1} = \max(-\infty, \log P_T(1 \mid 0) + \log P_E(b \mid 1) + f_{1,0})$$
$$= -3.730$$

Viterbi Algorithm: Example

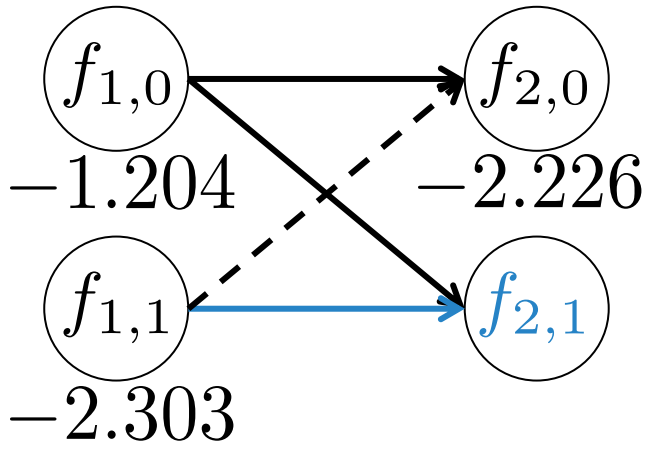
$y\backslash x$	a	b
0	0.6	0.4
1	0.2	**0.8**

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	**0.9**

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,1} = \max(-\infty, \log P_T(1 \mid 0) + \log P_E(b \mid 1) + f_{1,0})$$
$$= -3.730$$
$$f_{2,1} = \max(f_{2,1}, \log P_T(1 \mid 1) + \log P_E(b \mid 1) + f_{1,1})$$
$$= \max(-3.730, -2.631) = -2.631$$

Viterbi Algorithm: Example

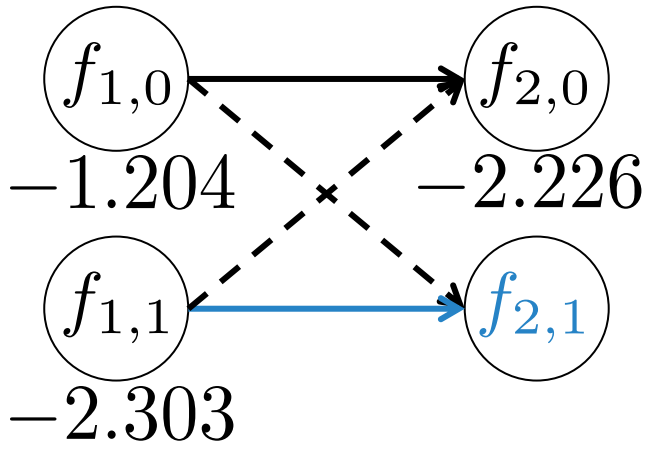
$y\backslash x$	a	b
0	0.6	0.4
1	0.2	**0.8**

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	**0.9**

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,1} = \max(-\infty, \log P_T(1 \mid 0) + \log P_E(b \mid 1) + f_{1,0})$$
$$= -3.730$$
$$f_{2,1} = \max(f_{2,1}, \log P_T(1 \mid 1) + \log P_E(b \mid 1) + f_{1,1})$$
$$= \max(-3.730, -2.631) = -2.631$$

Viterbi Algorithm: Example

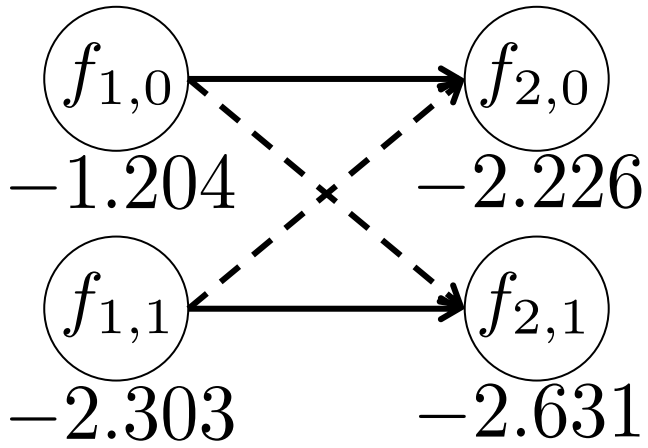
$y\backslash x$	a	b
0	0.6	0.4
1	0.2	0.8

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	**0.5**

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,0} + \log P(\langle \text{eos} \rangle \mid 0) = -2.919$$
$$f_{2,1} + \log P(\langle \text{eos} \rangle \mid 1) = -3.324$$

Viterbi Algorithm: Example

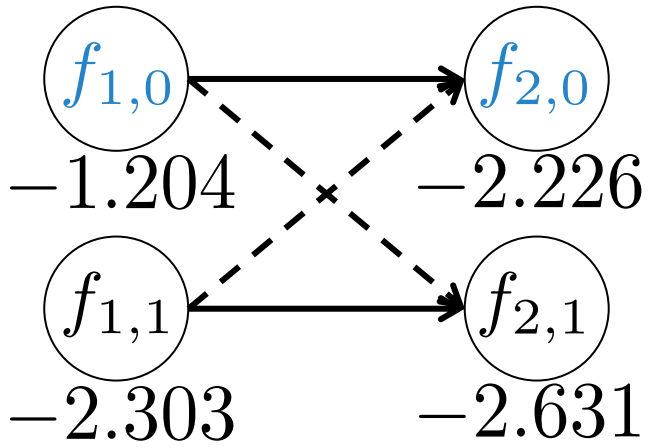
$y\backslash x$	a	b
0	0.6	0.4
1	0.2	0.8

$y_{t-1}\backslash y_t$	0	1
0	0.9	0.1
1	0.1	0.9

start	0	1
	0.5	0.5

$\langle eos \rangle$	0	1
	0.5	0.5

$$f_{k,y} : \max_{Y_k = y} \sum_{t=1}^{k} \log P_T(y_t \mid y_{t-1}) + \log P_E(x_t \mid y_t)$$



$$f_{2,0} + \log P(\langle \text{eos} \rangle \mid 0) = -2.919$$
$$f_{2,1} + \log P(\langle \text{eos} \rangle \mid 1) = -3.324$$

This Lecture

- Hidden Markov Models
  - Formulation and properties
  - Learning: estimate the parameters
  - Inference: finding the highest-scoring sequence of hidden variable values

- **Conditional Random Fields**

Conditional Random Fields

Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data

John Lafferty[†*] LAFFERTY@CS.CMU.EDU
Andrew McCallum[*†] MCCALLUM@WHIZBANG.COM
Fernando Pereira[*‡] FPEREIRA@WHIZBANG.COM
[*]WhizBang! Labs–Research, 4616 Henry Street, Pittsburgh, PA 15213 USA
[†]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA
[‡]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA

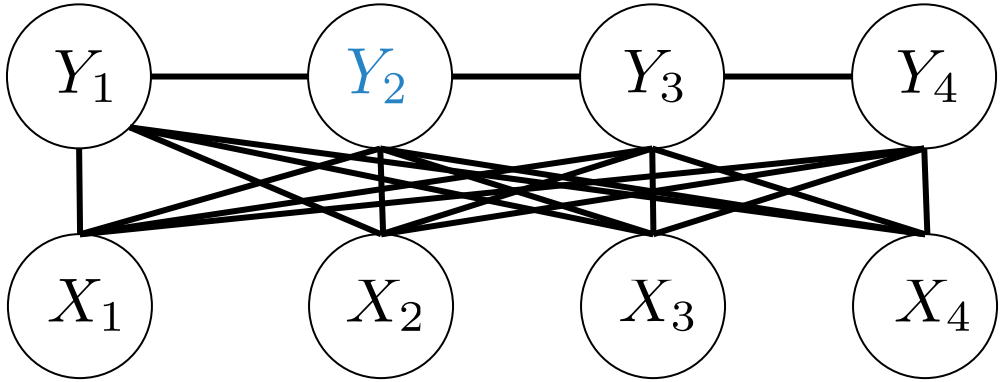
Abstract

We present *conditional random fields*, a framework for building probabilistic models to segment and label sequence data. Conditional random fields offer several advantages over hidden Markov models and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models. Conditional random fields also avoid a fundamental limitation of maximum entropy Markov models (MEMMs) and other discriminate the joint likelihood of training examples. To define a joint probability over observation and label sequences, a generative model needs to enumerate all possible observation sequences, typically requiring a representation in which observations are task-appropriate atomic entities, such as words or nucleotides. In particular, it is not practical to represent multiple interacting features or long-range dependencies of the observations, since the inference problem for such models is intractable.

This difficulty is one of the main motivations for looking at conditional models as an alternative. A conditional model

Conditional Random Fields

- Model a probability distribution with an undirected graph
- Variables are partitioned to two groups X and Y
- Models $P(Y \mid X)$
- Markov property: a variable in Y only depends on its neighbors



$$P(Y_2 \mid X, Y_1, Y_3, Y_4) = P(Y_2 \mid X, Y_1, Y_3)$$

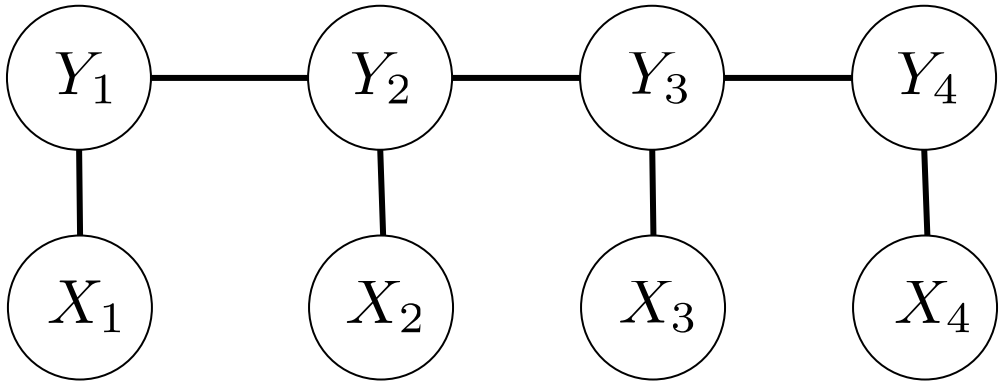
(Linear) Conditional Random Fields

- Model a probability distribution with an undirected graph

$$\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_i w_i f_i(\boldsymbol{x}, \boldsymbol{y})$$

$$P(\boldsymbol{y} \mid \boldsymbol{x}; \mathbf{w}) = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{\sum_{\boldsymbol{y}'} e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}'; \mathbf{w})}} = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{Z(\boldsymbol{x})}$$

- The feature function is usually defined among x_t, y_t, y_{t-1}



Conditional Random Fields: Learning

- (Stochastic) gradient descent

$$P(\boldsymbol{y} \mid \boldsymbol{x}; \mathbf{w}) = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{\sum_{\boldsymbol{y}'} e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}'; \mathbf{w})}} = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{Z(\boldsymbol{x})}$$

$$\mathcal{L}(\mathbf{w}; \left\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right\}) = -\sum_i \log P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}; \mathbf{w})$$

Conditional Random Fields: Learning

- (Stochastic) gradient descent

$$P(\boldsymbol{y} \mid \boldsymbol{x}; \mathbf{w}) = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{\sum_{\boldsymbol{y}'} e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}'; \mathbf{w})}} = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{Z(\boldsymbol{x})}$$

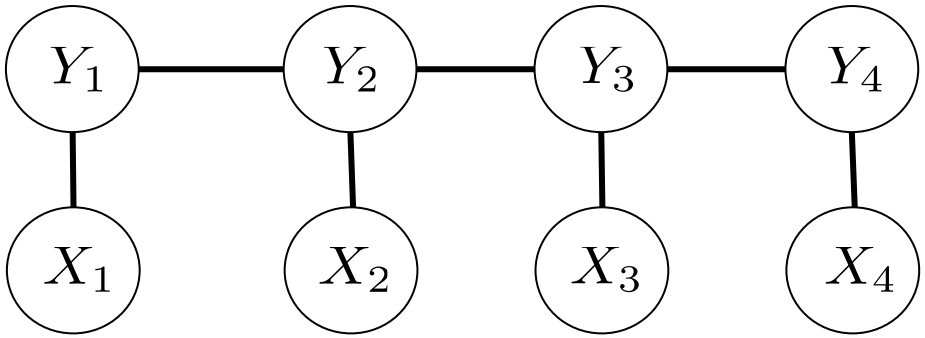
$$\mathcal{L}(\mathbf{w}; \left\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right\}) = -\sum_i \log P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}; \mathbf{w})$$

- The gradient can be efficiently calculated with dynamic programming (See Collins' notes for details)

Conditional Random Fields: Inference

$$P(\boldsymbol{y} \mid \boldsymbol{x}; \mathbf{w}) = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{\sum_{\boldsymbol{y}'} e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}'; \mathbf{w})}} = \frac{e^{\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})}}{Z(\boldsymbol{x})}$$

$$\text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_i w_i f_i(\boldsymbol{x}, \boldsymbol{y}) \qquad f_i(\boldsymbol{x}, \boldsymbol{y}) = \phi(x_t, y_{t-1}, y_t)$$



- Given **w**, find $\arg\max_{\boldsymbol{y}} P(\boldsymbol{y} \mid \boldsymbol{x})$
- Equivalent to finding

$$\arg\max_{\boldsymbol{y}} \text{score}(\boldsymbol{x}, \boldsymbol{y}; \mathbf{w})$$

- Viterbi algorithm!

Summary

- Hidden Markov Models
  - Formulation and properties
  - Learning: estimate the parameters
  - Inference: finding the highest-scoring sequence of hidden variable values

- Conditional Random Fields
  - Learning: gradient descent
  - Inference: Viterbi algorithm
  - Reading: Michael Collins' notes on CRF