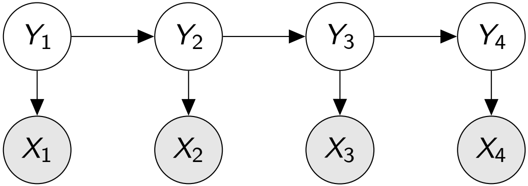
CS 784: Computational Linguistics
Lecture 12: Conditional Random Fields and Neural Sequence Labeling

Freda Shi

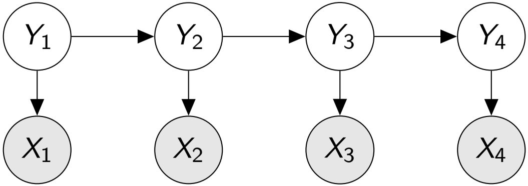
School of Computer Science, University of Waterloo
fhs@uwaterloo.ca

March 3, 2025

Drawbacks of HMMs



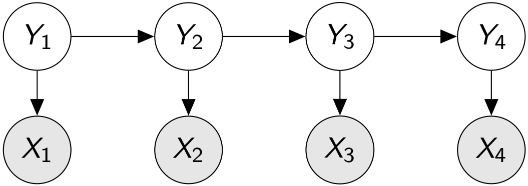
Drawbacks of HMMs



Each X_i and Y_i is a discrete random variable, which does not allow for rich feature representations.

For example, what if we'd like to consider morphological features of words?

Drawbacks of HMMs



Each X_i and Y_i is a discrete random variable, which does not allow for rich feature representations.

For example, what if we'd like to consider morphological features of words?

This motivates the use of **conditional random fields** (CRFs).

Conditional Random Fields

Laferty et al. (2001) introduced CRFs as a discriminative model for sequence labeling.

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

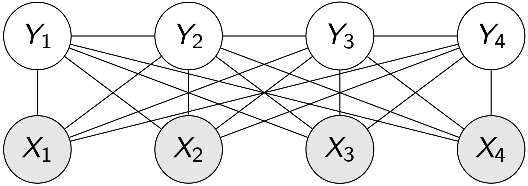
We present *conditional random fields*, a framework for building probabilistic models to segment and label sequence data. Conditional random fields offer several advantages over hidden Markov models and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models. Conditional random fields also avoid a fundamental limitation of maximum entropy Markov models (MEMMs) and other discrimi-

mize 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

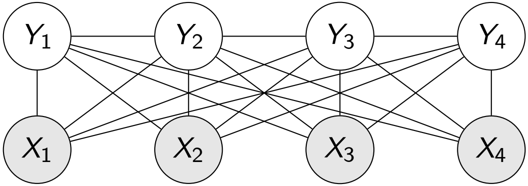
Conditional Random Fields

- Model the probability distribution $P(Y \mid X)$ with an **undirected graph**. Variables are partitioned into two sets: X (input) and Y (output).



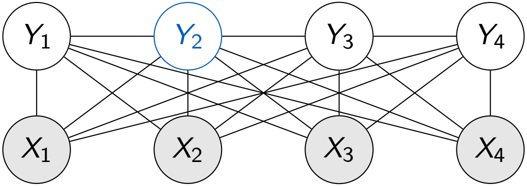
Conditional Random Fields

- Model the probability distribution $P(Y \mid X)$ with an **undirected graph**. Variables are partitioned into two sets: X (input) and Y (output).
- A special case of undirected probabilistic graphical models, which are also known as **Markov networks**.



Conditional Random Fields

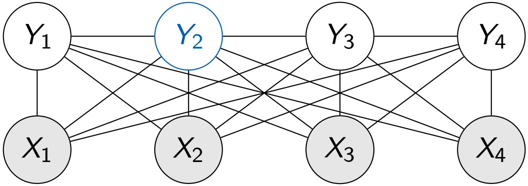
- Model the probability distribution $P(Y \mid X)$ with an **undirected graph**. Variables are partitioned into two sets: X (input) and Y (output).
- A special case of undirected probabilistic graphical models, which are also known as **Markov networks**.
  The **Markov property**: a random variable is conditionally independent of all others given its neighbors.



$$P(Y_2 \mid Y_1, Y_3, Y_4, X_{1:4}) =$$

Conditional Random Fields

- Model the probability distribution $P(Y \mid X)$ with an **undirected graph**. Variables are partitioned into two sets: X (input) and Y (output).
- A special case of undirected probabilistic graphical models, which are also known as **Markov networks**.
  The **Markov property**: a random variable is conditionally independent of all others given its neighbors.



$$P(Y_2 \mid Y_1, Y_3, Y_4, X_{1:4}) = P(Y_2 \mid Y_1, Y_3, X_{1:4})$$

(Linear) Conditional Random Fields

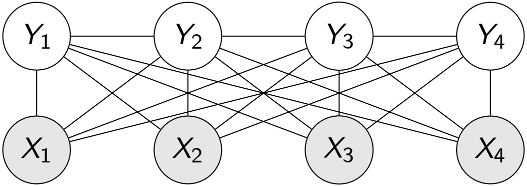
Recap: in a linear model, we score a input feature vector **x** with

$$\text{score}(\mathbf{x}; \mathbf{w}) = \sum_{i=1}^{n} w_i x_i.$$

(Linear) Conditional Random Fields

Recap: in a linear model, we score a input feature vector **x** with

$$\text{score}(\mathbf{x}; \mathbf{w}) = \sum_{i=1}^{n} w_i x_i.$$



In a linear CRF, we augment the linear model with **global features** $\mathbf{F}(X, Y)$ in the form of local feature sum:

$$F_k(X, Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i).$$

These features are to be used with the linear model to score the output sequence.

Example Features in Linear CRF

Each feature f_k is a function of the previous and current labels, the input sequence, and the current position

$$f_k(y_{i-1}, y_i, X, i).$$

Example Features in Linear CRF

Each feature f_k is a function of the previous and current labels, the input sequence, and the current position

$$f_k(y_{i-1}, y_i, X, i).$$

For example, in part-of-speech tagging, we might have features like

$$f(y_{i-1}, y_i, X, i) = \begin{cases} 1 & \text{if } y_{i-1} = \text{NOUN and } y_i = \text{VERB} \\ 0 & \text{otherwise} \end{cases}$$

Example Features in Linear CRF

Each feature f_k is a function of the previous and current labels, the input sequence, and the current position

$$f_k(y_{i-1}, y_i, X, i).$$

For example, in part-of-speech tagging, we might have features like

$$f(y_{i-1}, y_i, X, i) = \begin{cases} 1 & \text{if } y_{i-1} = \text{NOUN and } y_i = \text{VERB} \\ 0 & \text{otherwise} \end{cases}$$

$$f(y_{i-1}, y_i, X, i) = \begin{cases} 1 & \text{if } y_i = \text{DET and } x_{i+1} = \text{cat} \\ 0 & \text{otherwise} \end{cases}$$

Example Features in Linear CRF

Each feature f_k is a function of the previous and current labels, the input sequence, and the current position

$$f_k(y_{i-1}, y_i, X, i).$$

For example, in part-of-speech tagging, we might have features like

$$f(y_{i-1}, y_i, X, i) = \begin{cases} 1 & \text{if } y_{i-1} = \text{NOUN and } y_i = \text{VERB} \\ 0 & \text{otherwise} \end{cases}$$

$$f(y_{i-1}, y_i, X, i) = \begin{cases} 1 & \text{if } y_i = \text{DET and } x_{i+1} = \text{cat} \\ 0 & \text{otherwise} \end{cases}$$

For simplicity, we usually assume that the features are binary.

CRF: Formulation

A CRF defines a probability distribution over the output sequence Y given the input sequence X:

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X, Y)\right)$$
$$= \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)\right)$$

CRF: Formulation

A CRF defines a probability distribution over the output sequence Y given the input sequence X:

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X, Y)\right) = \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)\right)$$

The **partition function** $Z(X)$ is a normalization term that ensures the distribution sums to 1:

$$Z(X) = \sum_{Y'} \exp\left(\sum_{k=1}^{K} w_k F_k(X, Y')\right)$$

Inference for CRFs

Finding the most likely sequence of labels Y given the input X:

$$\arg\max_{Y} P(Y \mid X)$$

Inference for CRFs

Finding the most likely sequence of labels Y given the input X:

$$\arg\max_Y P(Y \mid X) = \arg\max_Y \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X, Y)\right)$$

Inference for CRFs

Finding the most likely sequence of labels Y given the input X:

$$\arg\max_Y P(Y \mid X) = \arg\max_Y \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X, Y)\right)$$

$$= \arg\max_Y \sum_{k=1}^{K} w_k F_k(X, Y)$$

$$= \arg\max_Y \sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

Inference for CRFs

Finding the most likely sequence of labels Y given the input X:

$$\begin{aligned}
\arg\max_{Y} P(Y \mid X) &= \arg\max_{Y} \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X, Y)\right) \\
&= \arg\max_{Y} \sum_{k=1}^{K} w_k F_k(X, Y) \\
&= \arg\max_{Y} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i) \\
&= \arg\max_{Y} \sum_{i=1}^{n} \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, X, i)
\end{aligned}$$

Inference for CRFs

Finding the most likely sequence of labels Y given the input X:

$$\begin{aligned}
\arg\max_{Y} P(Y \mid X) &= \arg\max_{Y} \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X, Y)\right) \\
&= \arg\max_{Y} \sum_{k=1}^{K} w_k F_k(X, Y) \\
&= \arg\max_{Y} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i) \\
&= \arg\max_{Y} \sum_{i=1}^{n} \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, X, i)
\end{aligned}$$

This is simply a variation of the Viterbi algorithm.

Hint: define $F[i, j]$ as the score of the best sequence ending at position i and Y_i taking value j.

Training CRFs

CRF training is essentially maximum (log) likelihood estimation:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{(X,Y)\in\mathcal{D}} \log P(Y \mid X; \mathbf{w})$$

Training CRFs

CRF training is essentially maximum (log) likelihood estimation:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{(X,Y)\in\mathcal{D}} \log P(Y \mid X; \mathbf{w})$$

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \sum_{(X,Y)\in\mathcal{D}} -\log P(Y \mid X; \mathbf{w}) \\ &= \sum_{(X,Y)\in\mathcal{D}} -\left(\sum_{k=1}^{K} w_k F_k(X, Y) - \log Z(X)\right)\end{aligned}$$

Training CRFs

CRF training is essentially maximum (log) likelihood estimation:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{(X,Y)\in\mathcal{D}} \log P(Y \mid X; \mathbf{w})$$

$$\mathcal{L}(\mathbf{w}) = \sum_{(X,Y)\in\mathcal{D}} -\log P(Y \mid X; \mathbf{w})$$

$$= \sum_{(X,Y)\in\mathcal{D}} -\left(\sum_{k=1}^{K} w_k F_k(X, Y) - \log Z(X)\right)$$

The gradient can be computed with the forward-backward algorithm, similarly to HMMs.

Sequence Labeling with Neural Networks

- Neural networks have been widely used for sequence labeling tasks.

Sequence Labeling with Neural Networks

- Neural networks have been widely used for sequence labeling tasks.
- Bi-LSTM + CRF was a popular architecture for sequence labeling.

Sequence Labeling with Neural Networks

- Neural networks have been widely used for sequence labeling tasks.
- Bi-LSTM + CRF was a popular architecture for sequence labeling.

However, pretrained neural feature + simple per-position classifier approach could give a competitive performance (Shi et al., 2021).

Intuition?

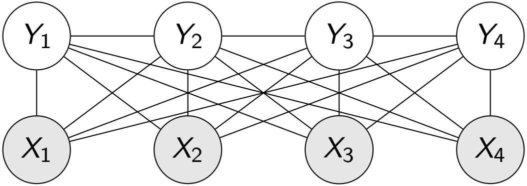
Sequence Labeling with Neural Networks

- Neural networks have been widely used for sequence labeling tasks.
- Bi-LSTM + CRF was a popular architecture for sequence labeling.

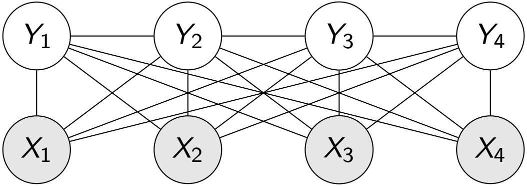
However, pretrained neural feature + simple per-position classifier approach could give a competitive performance (Shi et al., 2021).

Intuition? Powerful pretrained models already contain (almost) all information learned by graphical models.

The Neural CRF Layer



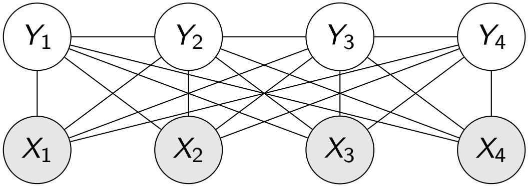
The Neural CRF Layer



The **emission score** at position i, $\text{e-score}(i) = \text{NN}(X) \in \mathbb{R}^C$, is computed by a per-position neural scorer (C: number of classes).

Note: not to be confused with the emission probability in HMMs.

The Neural CRF Layer

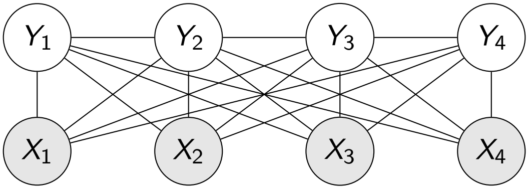


The **emission score** at position i, e-score$(i) = \text{NN}(X) \in \mathbb{R}^C$, is computed by a per-position neural scorer (C: number of classes).

Note: not to be confused with the emission probability in HMMs.

The **transition score** between Y_i and $Y_{i+1} (\in \mathbb{R}^{C \times C})$, is stored as parameters.

The Neural CRF Layer



The **emission score** at position i, $\text{e-score}(i) = \text{NN}(X) \in \mathbb{R}^C$, is computed by a per-position neural scorer (C: number of classes).

Note: not to be confused with the emission probability in HMMs.

The **transition score** between Y_i and $Y_{i+1} (\in \mathbb{R}^{C \times C})$, is stored as parameters.

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y_i} + \sum_{i=1}^{n-1} \text{t-score}(y_i, y_{i+1})\right)$$

The Neural CRF Layer: Partition Function

$Z(X)$ can be computed with the forward algorithm:

$$Z(X) = \sum_{Y'} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y'_i} + \sum_{i=1}^{n-1} \text{t-score}(y'_i, y'_{i+1}) \right)$$

The Neural CRF Layer: Partition Function

$Z(X)$ can be computed with the forward algorithm:

$$
\begin{aligned}
Z(X) &= \sum_{Y'} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y'_i} + \sum_{i=1}^{n-1} \text{t-score}(y'_i, y'_{i+1})\right) \\
&= \sum_{y_1} \dots \sum_{y_n} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y_i} + \sum_{i=1}^{n-1} \text{t-score}(y_i, y_{i+1})\right)
\end{aligned}
$$

The Neural CRF Layer: Partition Function

$Z(X)$ can be computed with the forward algorithm:

$$Z(X) = \sum_{Y'} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y'_i} + \sum_{i=1}^{n-1} \text{t-score}(y'_i, y'_{i+1})\right)$$

$$= \sum_{y_1} \dots \sum_{y_n} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y_i} + \sum_{i=1}^{n-1} \text{t-score}(y_i, y_{i+1})\right)$$

$$= \sum_{y_1} \exp\left(\text{e-score}(1)_{y_1}\right) \cdot$$

$$\left(\sum_{y_2} \dots \sum_{y_n} \exp\left(\sum_{i=2}^{n} \text{e-score}(i)_{y_i} + \sum_{i=1}^{n-1} \text{t-score}(y_i, y_{i+1})\right)\right)$$

The Neural CRF Layer: Partition Function

$Z(X)$ can be computed with the forward algorithm:

$$
\begin{aligned}
Z(X) &= \sum_{Y'} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y'_i} + \sum_{i=1}^{n-1} \text{t-score}(y'_i, y'_{i+1})\right) \\
&= \sum_{y_1} \ldots \sum_{y_n} \exp\left(\sum_{i=1}^{n} \text{e-score}(i)_{y_i} + \sum_{i=1}^{n-1} \text{t-score}(y_i, y_{i+1})\right) \\
&= \sum_{y_1} \exp\left(\text{e-score}(1)_{y_1}\right) \cdot \\
&\quad \left(\sum_{y_2} \ldots \sum_{y_n} \exp\left(\sum_{i=2}^{n} \text{e-score}(i)_{y_i} + \sum_{i=1}^{n-1} \text{t-score}(y_i, y_{i+1})\right)\right)
\end{aligned}
$$

Training objective: maximize the log-likelihood of training data.

With automatic differentiation, compute $P(Y \mid X)$ is everything!

Next

Syntax and parsing