CS 6501 Natural Language Processing

Viterbi Decoding, CRFs

Yangfeng Ji

September 24, 2019

Department of Computer Science
University of Virginia

Overview

1. Viterbi Decoding
2. Review: HMMs and Logistic Regression
3. Conditional Random Fields
4. Inference

Viterbi Decoding

Factorization

Factorization of $p(\boldsymbol{x}, \boldsymbol{y})$, given the first-order Markov property

$$
\begin{aligned}
p(\boldsymbol{x}, \boldsymbol{y}) &= \prod_{i=1}^{T} \Big\{ p(y_i \mid y_{i-1}) p(x_i \mid y_i) \Big\} p(\blacksquare \mid y_T) \qquad (1) \\
&= \prod_{i=1}^{T} \Big\{ p(x_i, y_i \mid y_{i-1}) \Big\} p(\blacksquare \mid y_T) \qquad (2)
\end{aligned}
$$

where $y_0 = \square$.

Factorization

Factorization of $p(\boldsymbol{x}, \boldsymbol{y})$, given the first-order Markov property

$$p(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{T} \Big\{ p(y_i \mid y_{i-1}) p(x_i \mid y_i) \Big\} p(\blacksquare \mid y_T) \tag{1}$$

$$= \prod_{i=1}^{T} \Big\{ p(x_i, y_i \mid y_{i-1}) \Big\} p(\blacksquare \mid y_T) \tag{2}$$

where $y_0 = \square$.

Or, in log space

$$\log p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \Big\{ \log p(y_i \mid y_{i-1}) + \log p(x_i \mid y_i) \Big\} + \log p(\blacksquare \mid y_T) \tag{3}$$

$$= \sum_{i=1}^{T} \Big\{ \log p(x_i, y_i \mid y_{i-1}) \Big\} + \log p(\blacksquare \mid y_T) \tag{4}$$

Decoding $\boldsymbol{y}$

For a given sentence, $\boldsymbol{x}$ is fixed, therefore

$$\begin{aligned}
\hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}) && (5)\\
&= \operatorname*{argmax}_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y}) && (6)\\
&= \operatorname*{argmax}_{\boldsymbol{y}} \log p(\boldsymbol{x}, \boldsymbol{y}) && (7)
\end{aligned}$$

Decoding $\boldsymbol{y}$

For a given sentence, $\boldsymbol{x}$ is fixed, therefore

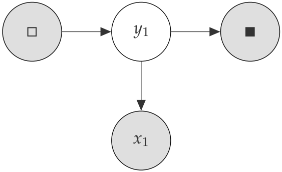
$$\begin{aligned}
\hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}) && (5)\\
&= \operatorname*{argmax}_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y}) && (6)\\
&= \operatorname*{argmax}_{\boldsymbol{y}} \log p(\boldsymbol{x}, \boldsymbol{y}) && (7)
\end{aligned}$$

Consider a very special case, where $T = 1$

$$\log p(\boldsymbol{x}, \boldsymbol{y}) = \log p(y_1 \mid \square) + \log p(x_1 \mid y_1) + \log(\blacksquare \mid y_1) \quad (8)$$

Graphical Model

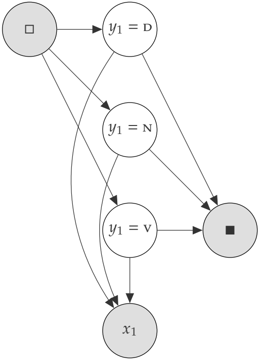
$$\log p(\boldsymbol{x}, \boldsymbol{y}) = \log p(y_1 \mid \square) + \log p(x_1 \mid y_1) + \log(\blacksquare \mid y_1) \tag{9}$$



$$\log p(\boldsymbol{x}, \boldsymbol{y}) = \log p(y_1 \mid \square) {+} \log p(x_1 \mid y_1) {+} \log(\blacksquare \mid y_1) \tag{10}$$

Trellis

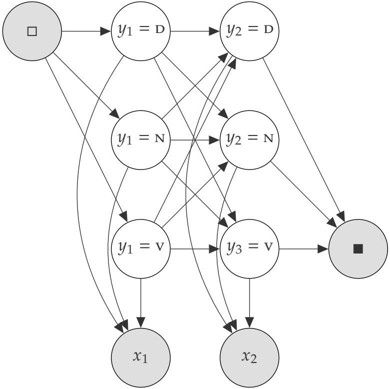
Assume the sample space of y_i is {D,N,V}, then the previous graphical model can be extended as a trellis representation as



Try every value of y_1, then we can find the optimal

When $T = 2$

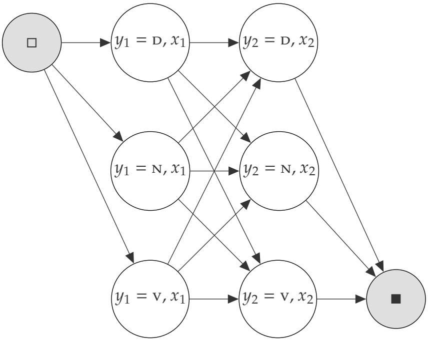
With the same problem setup and now $T = 2$



To simplify the graph notations ...

When $T = 2$

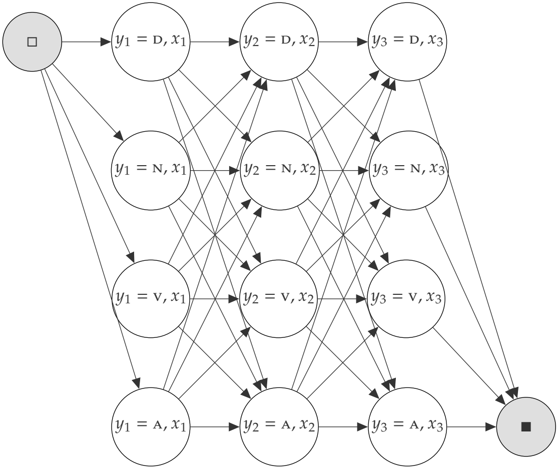
Absorbing $p(x_i \mid y_1)$ into each node of y_i, we have



$$\begin{aligned}
\log p(\boldsymbol{x}, \boldsymbol{y}) &= \sum \Big\{ \log p(y_i \mid y_{i-1}) + \log p(x_i \mid y_i) \Big\} + \log p(\blacksquare \mid y_T) \\
&= \sum \Big\{ \log p(x_i, y_i \mid y_{i-1}) \Big\} + \log p(\blacksquare \mid y_T) \qquad (11)
\end{aligned}$$

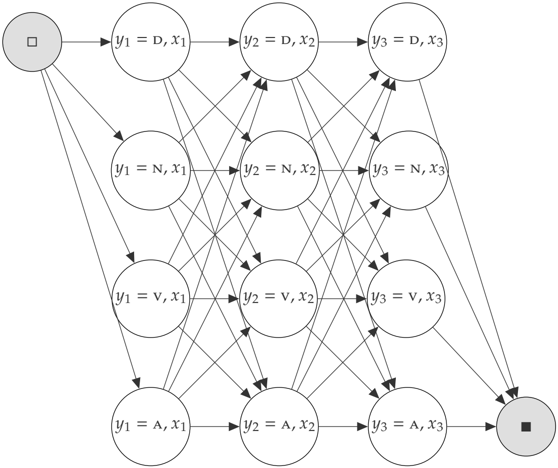
When $T = 3$

Consider a little more compliciated case, where $T = 3$ and each y_i has four different states {D,N,V,A}



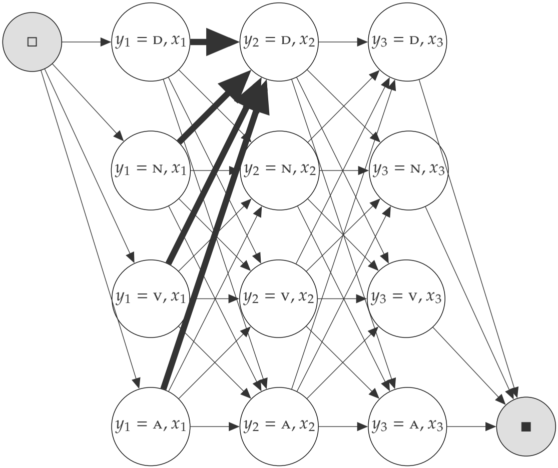
Forward computation: on y_t

Forward computation: at each timestamp t, for each value y_t, find the best path that leads to y_t



Forward computation: on y_t

Forward computation: at each timestamp t, for each value y_t, find the best path that leads to y_t



Viterbi Variable

Viterbi variable $v(y_t)$ is the best value reaching to y_t

$$v(y_t) = \max_{\boldsymbol{y}_{<t}} \log p(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t, y_t)$$

Viterbi Variable

Viterbi variable $v(y_t)$ is the best value reaching to y_t

$$
\begin{aligned}
v(y_t) &= \max_{\boldsymbol{y}_{<t}} \log p(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t, y_t) \\
&= \max_{y_{t-1}} \left\{ \max_{\boldsymbol{y}_{<t-1}} \log p(\boldsymbol{x}_{<t-1}, \boldsymbol{y}_{<t-1}, x_{t-1}, y_{t-1}, x_t, y_t) \right\}
\end{aligned}
$$

Viterbi Variable

Viterbi variable $v(y_t)$ is the best value reaching to y_t

$$
\begin{aligned}
v(y_t) &= \max_{\boldsymbol{y}_{<t}} \log p(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t, y_t) \\
&= \max_{y_{t-1}} \left\{ \max_{\boldsymbol{y}_{<t-1}} \log p(\boldsymbol{x}_{<t-1}, \boldsymbol{y}_{<t-1}, x_{t-1}, y_{t-1}, x_t, y_t) \right\} \\
&= \max_{y_{t-1}} \left\{ \max_{\boldsymbol{y}_{<t-1}} \log p(\boldsymbol{x}_{<t-1}, \boldsymbol{y}_{<t-1}, x_{t-1}, y_{t-1}) \right. \\
&\qquad \left. + \log p(x_t, y_t \mid y_{t-1}) \right\} \quad \text{Markov property}
\end{aligned}
$$

Viterbi Variable

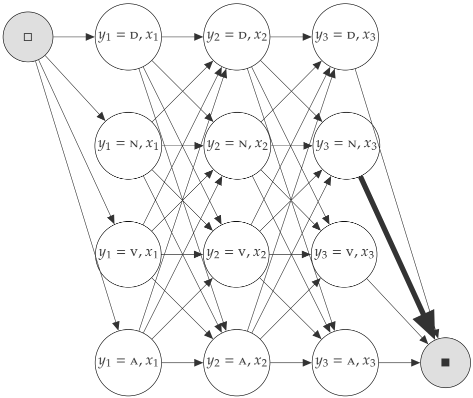
Viterbi variable $v(y_t)$ is the best value reaching to y_t

$$
\begin{aligned}
v(y_t) &= \max_{\boldsymbol{y}_{<t}} \log p(\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t}, x_t, y_t) \\
&= \max_{y_{t-1}} \left\{ \max_{\boldsymbol{y}_{<t-1}} \log p(\boldsymbol{x}_{<t-1}, \boldsymbol{y}_{<t-1}, x_{t-1}, y_{t-1}, x_t, y_t) \right\} \\
&= \max_{y_{t-1}} \left\{ \max_{\boldsymbol{y}_{<t-1}} \log p(\boldsymbol{x}_{<t-1}, \boldsymbol{y}_{<t-1}, x_{t-1}, y_{t-1}) \right. \\
&\qquad \left. + \log p(x_t, y_t \mid y_{t-1}) \right\} \quad \text{Markov property} \\
&= \max_{y_{t-1}} \{ v(y_{t-1}) + \log p(x_t, y_t \mid y_{t-1}) \}
\end{aligned}
$$

A Special Case

When $t = T + 1$

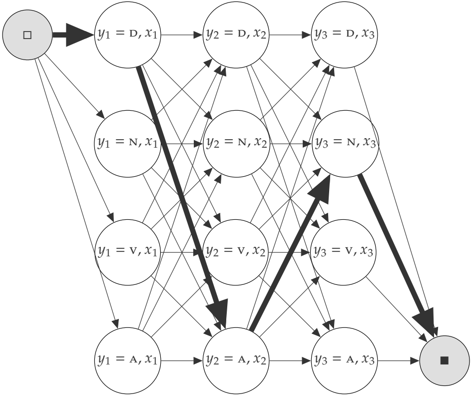
- $v(\blacksquare)$ is the best value reaching to $\blacksquare$



A Special Case

When $t = T + 1$

- $v(\blacksquare)$ is the best value reaching to $\blacksquare$



Basic Idea of Decoding

$$v(y_t) = \max_{y_{t-1}} \{v(y_{t-1}) + \log p(x_t, y_t \mid y_{t-1})\} \tag{12}$$

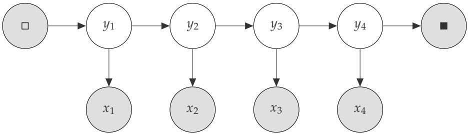
- For each y_t, find the best value of y_{t-1} that leads to y_t
- After reaching to ■, trace back to find the best path on trellis

Review: HMMs and Logistic Regression

Hidden Markov Models

$$p(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1} \left\{ p(y_i | y_{i-1}) P(x_i | y_i) \right\} \qquad (13)$$

Graphical model



- $\boldsymbol{x}$: observation (e.g., sentences)
- $\boldsymbol{y}$: hidden variables (e.g., POS sequences)

Generative Models

$$p(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y}) \tag{14}$$

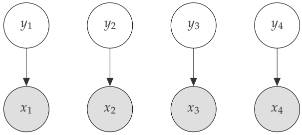
Generative Models

$$p(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}|\boldsymbol{y}) \cdot P(\boldsymbol{y}) \tag{14}$$

Factorization

$$p(\boldsymbol{x}|\boldsymbol{y}) = \prod_{i=1} \underbrace{p(x_i|y_i)}_{\text{Emission probability}} \tag{15}$$

Graphical model



Label Classification

Example

$\boldsymbol{x}$	x_1 Teacher	x_2 Strikes	x_3 Idle	x_4 Children
$\boldsymbol{y}$	y_1	y_2	y_3	y_4
	NOUN	NOUN	VERB	NOUN
	NOUN	VERB	ADJ	NOUN

Limitations

- ✓ No constraint from the previous POS tag
  - Solution: sequence labeling (e.g., hidden Markove models, conditional random fields)
- No information from the surrounding words
  - Solution: conditional random fields

Discriminative Models: Logistic Regression

$$P(y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta}_y^\top \boldsymbol{f}(\boldsymbol{x}))}{\sum_{y' \in \mathcal{Y}} \exp(\boldsymbol{\theta}_{y'}^\top \boldsymbol{f}(\boldsymbol{x}))} \tag{16}$$

where

- y is a random variable (scalar)
- $\boldsymbol{f}(\boldsymbol{x})$ is a feature function
- $\boldsymbol{\theta}_y$ is the classification weight associated with label y

Discriminative Models: Logistic Regression

$$P(y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta}_y^\top \boldsymbol{f}(\boldsymbol{x}))}{\sum_{y' \in \mathcal{Y}} \exp(\boldsymbol{\theta}_{y'}^\top \boldsymbol{f}(\boldsymbol{x}))} \tag{16}$$

where

- y is a random variable (scalar)
- $\boldsymbol{f}(\boldsymbol{x})$ is a feature function
- $\boldsymbol{\theta}_y$ is the classification weight associated with label y

Question

What if $\boldsymbol{y}$ is a sequence of random variables?

Conditional Random Fields

Logistic Regression

A direct application of logistic regression:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}'))} \tag{17}$$

Huge $\mathcal{Y}^T$ causes the problems on

- decoding $\text{argmax}_{\boldsymbol{y}' \in \mathcal{Y}^T} p(\boldsymbol{y}'|\boldsymbol{x})$

Logistic Regression

A direct application of logistic regression:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \underbrace{\frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}'))}}_{\text{partition function } Z} \tag{17}$$

Huge $\mathcal{Y}^T$ causes the problems on

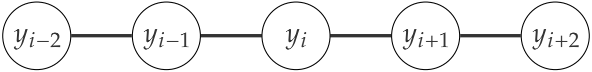
- decoding $\operatorname{argmax}_{\boldsymbol{y}' \in \mathcal{Y}^T} p(\boldsymbol{y}'|\boldsymbol{x})$
- computing the partition function with $|\mathcal{Y}^T| = K^T$ possible values

Markov Property

Global feature function:

$$f(x, y) \tag{18}$$

Markov assumption:



- Conditional independence
- Factorization over cliques

Decomposition of $f(x, y)$

$$f(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \underbrace{f_i(x_i, y_i, y_{i-1})}_{\text{local feature function}} \tag{19}$$

- i: the position to be tagged
- $y_i \in \mathcal{Y}$: POS tag at position i
- $y_{i-1} \in \mathcal{Y}$: POS tag at position $i-1$
- x_i: observation (word) at position i
- $f_i(x_i, y_i, y_{i-1})$ captures the dependency of (y_{i-1}, y_i) and (x_i, y_i)

Local Feature Function: Example

- standard features

[Lafferty et al., 2001]

Local Feature Function: Example

- standard features
- whether a spelling begins with upper case letter,
  - `IBM`, `Virginia`: PROPER NOUN

[Lafferty et al., 2001]

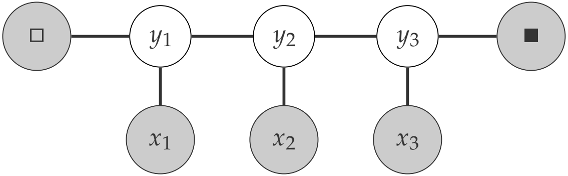
Local Feature Function: Example

- standard features
- whether a spelling begins with upper case letter,
  - `IBM`, `Virginia`: PROPER NOUN
- whether it ends in one of the following suffixes:
  - `-ies` e.g., `parties`: PROPER NOUN, PLURAL
  - `-ly` e.g., `extremely`, `loudly`: ADVERB
  - `-ing` e.g., : VERB, GERUND OR PRESENT PARTICIPLE
  - ...

[Lafferty et al., 2001]

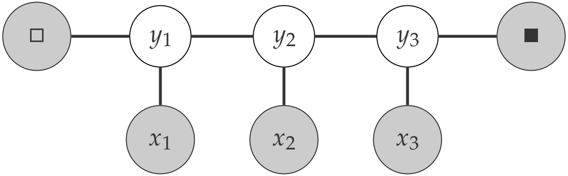
Graphical Model Representation

Conditional Random Fields:

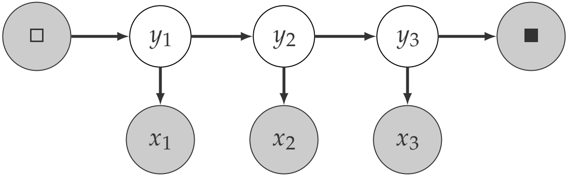


Graphical Model Representation

Conditional Random Fields:



Hidden Markov Models:



Inference

Decode $P(\boldsymbol{y}|\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \boldsymbol{f}_i(x_i, y_i, y_{i-1}) \tag{20}$$

Decode $P(\boldsymbol{y}|\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \boldsymbol{f}_i(x_i, y_i, y_{i-1}) \tag{20}$$

$$\begin{aligned}
\operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} P(\boldsymbol{y}|\boldsymbol{x}) &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}'))} \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}))
\end{aligned}$$

Decode $P(\boldsymbol{y}|\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \boldsymbol{f}_i(x_i, y_i, y_{i-1}) \tag{20}$$

$$\begin{aligned}
\operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} P(\boldsymbol{y}|\boldsymbol{x}) &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}'))} \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})) \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})
\end{aligned}$$

Decode $P(\boldsymbol{y}|\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \boldsymbol{f}_i(x_i, y_i, y_{i-1}) \tag{20}$$

$$\begin{aligned}
\operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} P(\boldsymbol{y}|\boldsymbol{x}) &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}'))} \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})) \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \boldsymbol{\theta}^\top \sum_{i=1}^{T} \boldsymbol{f}_i(x_i, y_i, y_{i-1})
\end{aligned}$$

Decode $P(\boldsymbol{y}|\boldsymbol{x})$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \boldsymbol{f}_i(x_i, y_i, y_{i-1}) \tag{20}$$

$$\begin{aligned}
\operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} P(\boldsymbol{y}|\boldsymbol{x}) &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \frac{\exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}))}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}'))} \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \exp(\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})) \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \boldsymbol{\theta}^\top \sum_{i=1}^{T} \boldsymbol{f}_i(x_i, y_i, y_{i-1}) \\
&= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^T} \sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{f}_i(x_i, y_i, y_{i-1})
\end{aligned}$$

Factorization

Factorize $\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})$ with respect to timestep i

$$
\begin{aligned}
\sum_{i=1}^{T} \boldsymbol{\theta}^\top \boldsymbol{f}_i(x_i, y_i, y_{i-1}) = & \underbrace{\sum_{j \le i-1} \boldsymbol{\theta}^\top \boldsymbol{f}_j(x_j, y_j, y_{j-1})} \\
& + \underbrace{\boldsymbol{\theta}^\top \boldsymbol{f}_i(x_i, y_i, y_{i-1})}_{\text{present}} \\
& + \underbrace{\sum_{k \ge i+1} \boldsymbol{\theta}^\top \boldsymbol{f}_k(x_k, y_k, y_{k-1})}_{\text{future}}
\end{aligned}
\tag{21}
$$

Viterbi Algorithm

$$s_i(k, k') = \boldsymbol{\theta}^\top \boldsymbol{f}_i(x_i, y_i = k, y_{i=1} = k')$$

Algorithm 11 The Viterbi algorithm. Each $s_m(k, k')$ is a local score for tag $y_m = k$ and $y_{m-1} = k'$.

for $k \in \{0, \ldots K\}$ **do**
  $v_1(k) = s_1(k, \Diamond)$
for $m \in \{2, \ldots, M\}$ **do**
  for $k \in \{0, \ldots, K\}$ **do**
    $v_m(k) = \max_{k'} s_m(k, k') + v_{m-1}(k')$
    $b_m(k) = \operatorname{argmax}_{k'} s_m(k, k') + v_{m-1}(k')$
$y_M = \operatorname{argmax}_k s_{M+1}(\blacklozenge, k) + v_M(k)$
for $m \in \{M - 1, \ldots 1\}$ **do**
  $y_m = b_m(y_{m+1})$
return $\boldsymbol{y}_{1:M}$

[Eisenstein, 2018]

Reference

Eisenstein, J. (2018).
Natural Language Processing.
MIT Press.

Lafferty, J., McCallum, A., and Pereira, F. (2001).
Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
In *ICML*.