Natural Language Processing (CSE 490U): Sequence Models (I)

Noah Smith

University of Washington
nasmith@cs.washington.edu

January 25, 2017

Linguistic Analysis: Overview

Every linguistic analyzer is comprised of:

1. Theoretical motivation from linguistics and/or the text domain
2. An algorithm that maps $\mathcal{V}^\dagger$ to some output space $\mathcal{Y}$.
3. An implementation of the algorithm
   - Once upon a time: rule systems and crafted rules
   - Most common now: supervised learning from annotated data
   - Frontier: less supervision (semi-, un-, reinforcement, distant, ...)

Sequence Labeling

After text classification ($\mathcal{V}^\dagger \rightarrow \mathcal{L}$), the next simplest type of output is a **sequence labeling**.

$$\langle x_1, x_2, \ldots, x_\ell \rangle \mapsto \langle y_1, y_2, \ldots, y_\ell \rangle$$
$$\boldsymbol{x} \mapsto \boldsymbol{y}$$

Every word gets a label in $\mathcal{L}$.
Example problems:

- part-of-speech tagging (Church, 1988)
- spelling correction (Kernighan et al., 1990)
- word alignment (Vogel et al., 1996)
- named-entity recognition (Bikel et al., 1999)
- compression (Conroy and O'Leary, 2001)

The Simplest Sequence Labeler: "Local" Classifier

Define features of a labeled word in context: $\boldsymbol{\phi}(\boldsymbol{x}, i, y)$.

Train a classifier, e.g.,

$$\hat{y}_i = \operatorname*{argmax}_{y \in \mathcal{L}} s(\boldsymbol{x}, i, y)$$
$$\overset{\text{linear}}{=} \operatorname*{argmax}_{y \in \mathcal{L}} \mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y)$$

Decide the label for each word independently.

The Simplest Sequence Labeler: "Local" Classifier

Define features of a labeled word in context: $\boldsymbol{\phi}(\boldsymbol{x}, i, y)$.

Train a classifier, e.g.,

$$\hat{y}_i = \operatorname*{argmax}_{y \in \mathcal{L}} s(\boldsymbol{x}, i, y)$$
$$\overset{\text{linear}}{=} \operatorname*{argmax}_{y \in \mathcal{L}} \mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y)$$

Decide the label for each word independently.

Sometimes this works!

The Simplest Sequence Labeler: "Local" Classifier

Define features of a labeled word in context: $\boldsymbol{\phi}(\boldsymbol{x}, i, y)$.

Train a classifier, e.g.,

$$\hat{y}_i = \operatorname*{argmax}_{y \in \mathcal{L}} s(\boldsymbol{x}, i, y)$$
$$\overset{\text{linear}}{=} \operatorname*{argmax}_{y \in \mathcal{L}} \mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y)$$

Decide the label for each word independently.

Sometimes this works!

We can do better when there are predictable relationships between Y_i and Y_{i+1}.

Generative Sequence Labeling: Hidden Markov Models

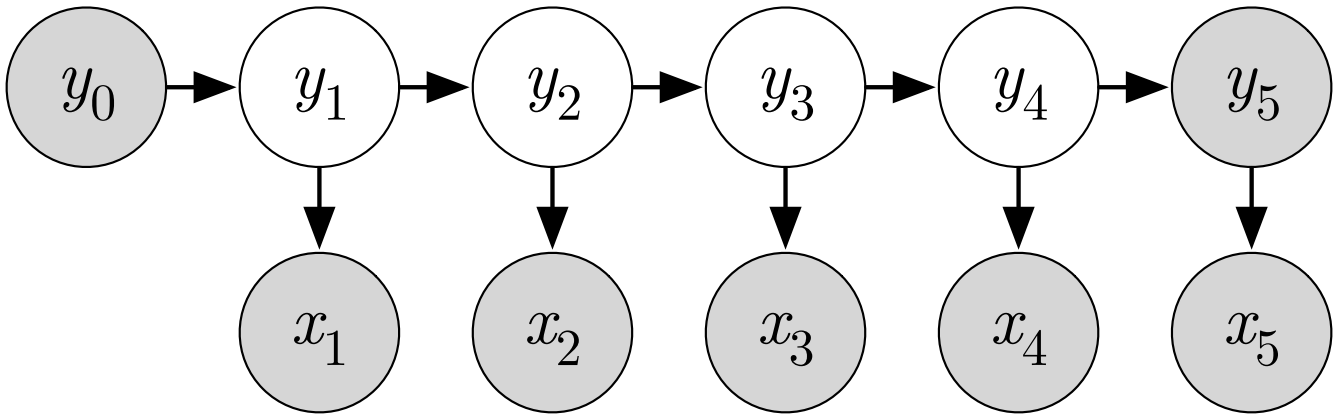
$$p(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{\ell+1} p(x_i \mid y_i) \cdot p(y_i \mid y_{i-1})$$

For each state/label $y \in \mathcal{L}$:

- $p(X_i \mid Y_i = y)$ is the "emission" distribution for y
- $p(Y_i \mid Y_{i-1} = y)$ is called the "transition" distribution for y

Assume Y_0 is always a start state and $Y_{\ell+1}$ is always a stop state; $x_{\ell+1}$ is always the stop symbol.

Graphical Reprsentation of Hidden Markov Models



Note: handling of beginning and end of sequence is a bit different than before. Last x is known since $p(\bigcirc \mid \bigcirc) = 1$.

Structured vs. Not

Each of these has an advantage over the other:

- The HMM lets the different labels "interact."
- The local classifier makes all of $\boldsymbol{x}$ available for every decision.

Prediction with HMMs

The classical HMM tells us to choose:

$$\operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} \prod_{i=1}^{\ell+1} p(x_i, \mid y_i) \cdot p(y_i \mid y_{i-1})$$

How to optimize over $|\mathcal{L}|^{\ell}$ choices without explicit enumeration?

Prediction with HMMs

The classical HMM tells us to choose:

$$\operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} \prod_{i=1}^{\ell+1} p(x_i, \mid y_i) \cdot p(y_i \mid y_{i-1})$$

How to optimize over $|\mathcal{L}|^{\ell}$ choices without explicit enumeration?

Key: exploit the conditional independence assumptions:

$$Y_i \perp \boldsymbol{Y}_{1:i-2} \mid Y_{i-1}$$
$$Y_i \perp \boldsymbol{Y}_{i+2:\ell} \mid Y_{i+1}$$

Part-of-Speech Tagging Example

	I	suspect	the	present	forecast	is	pessimistic	.
noun	•	•	•	•	•	•		
adj.		•		•	•		•	
adv.				•				
verb		•		•	•	•		
num.	•							
det.			•					
punc.								•

With this very simple tag set, $7^8 = 5.7$ million labelings.
(Even restricting to the possibilities above, 288 labelings.)

Two Obvious Solutions

Brute force: Enumerate all solutions, score them, pick the best.

Greedy: Pick each $\hat{y}_i$ according to:

$$\hat{y}_i = \operatorname*{argmax}_{y \in \mathcal{L}} p(y \mid \hat{y}_{i-1}) \cdot p(x_i \mid y)$$

What's wrong with these?

Two Obvious Solutions

Brute force: Enumerate all solutions, score them, pick the best.

Greedy: Pick each $\hat{y}_i$ according to:

$$\hat{y}_i = \underset{y \in \mathcal{L}}{\operatorname{argmax}}\, p(y \mid \hat{y}_{i-1}) \cdot p(x_i \mid y)$$

What's wrong with these?

Consider: "the horse raced past the barn fell"

Conditional Independence

We can get an exact solution in polynomial time!

$$Y_i \perp \boldsymbol{Y}_{1:i-2} \mid Y_{i-1}$$
$$Y_i \perp \boldsymbol{Y}_{i+2:\ell} \mid Y_{i+1}$$

Given the adjacent labels to Y_i, others do not matter.

Let's start at the last position, ℓ ...

Chart Data Structure

	x_1	x_2	$\ldots$	x_ℓ
y				
y'				
$\vdots$				
y^{last}				

High-Level View of Viterbi

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!

$$p(Y_\ell = y \mid \boldsymbol{x}, \boldsymbol{y}_{1:(\ell-1)}) = p\left(Y_\ell = y \,\middle|\, \begin{array}{l} X_\ell = x_\ell, \\ Y_{\ell-1} = y_{\ell-1}, \\ Y_{\ell+1} = \bigcirc \end{array}\right)$$

$$= \frac{p(Y_\ell = y, X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}{p(X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}$$

$$\propto p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot p(y \mid y_{\ell-1})$$

High-Level View of Viterbi

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!

$$
\begin{aligned}
p(Y_\ell = y \mid \boldsymbol{x}, \boldsymbol{y}_{1:(\ell-1)}) &= p\left(Y_\ell = y \,\middle|\, \begin{array}{l} X_\ell = x_\ell, \\ Y_{\ell-1} = y_{\ell-1}, \\ Y_{\ell+1} = \bigcirc \end{array}\right) \\
&= \frac{p(Y_\ell = y, X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}{p(X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)} \\
&\propto p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot p(y \mid y_{\ell-1})
\end{aligned}
$$

- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ would be easy.

High-Level View of Viterbi

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!

$$
\begin{aligned}
p(Y_\ell = y \mid \boldsymbol{x}, \boldsymbol{y}_{1:(\ell-1)}) &= p\left(Y_\ell = y \,\middle|\, \begin{array}{l} X_\ell = x_\ell, \\ Y_{\ell-1} = y_{\ell-1}, \\ Y_{\ell+1} = \bigcirc \end{array}\right) \\
&= \frac{p(Y_\ell = y, X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}{p(X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)} \\
&\propto p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot p(y \mid y_{\ell-1})
\end{aligned}
$$

- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ would be easy.
- Idea: for each position i, calculate the score of the best label prefix $\boldsymbol{y}_{1:i}$ ending in each possible value for Y_i.

High-Level View of Viterbi

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!

$$p(Y_\ell = y \mid \boldsymbol{x}, \boldsymbol{y}_{1:(\ell-1)}) = p\left(Y_\ell = y \,\middle|\, \begin{array}{l} X_\ell = x_\ell, \\ Y_{\ell-1} = y_{\ell-1}, \\ Y_{\ell+1} = \bigcirc \end{array}\right)$$

$$= \frac{p(Y_\ell = y, X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}{p(X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}$$

$$\propto p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot p(y \mid y_{\ell-1})$$

- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ would be easy.
- Idea: for each position i, calculate the score of the best label prefix $\boldsymbol{y}_{1:i}$ ending in each possible value for Y_i.
- With a little bookkeeping, we can then trace backwards and recover the best label sequence.

Recurrence

First, think about the *score* of the best sequence.

Let $s_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$s_\ell(y) = p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

Recurrence

First, think about the *score* of the best sequence.

Let $s_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$s_\ell(y) = p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$s_{\ell-1}(y) = p(x_{\ell-1} \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-2}(y')}$$

Recurrence

First, think about the *score* of the best sequence.

Let $s_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$s_\ell(y) = p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$s_{\ell-1}(y) = p(x_{\ell-1} \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-2}(y')}$$

$$s_{\ell-2}(y) = p(x_{\ell-2} \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-3}(y')}$$

Recurrence

First, think about the *score* of the best sequence.

Let $s_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$s_\ell(y) = p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$s_{\ell-1}(y) = p(x_{\ell-1} \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-2}(y')}$$

$$s_{\ell-2}(y) = p(x_{\ell-2} \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-3}(y')}$$

$$\vdots$$

$$s_i(y) = p(x_i \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{i-1}(y')}$$

Recurrence

First, think about the *score* of the best sequence.

Let $s_i(y)$ be the score of the best label sequence for $\boldsymbol{x}_{1:i}$ that ends in y. It is defined recursively:

$$s_\ell(y) = p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$s_{\ell-1}(y) = p(x_{\ell-1} \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-2}(y')}$$

$$s_{\ell-2}(y) = p(x_{\ell-2} \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-3}(y')}$$

$$\vdots$$

$$s_i(y) = p(x_i \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{i-1}(y')}$$

$$\vdots$$

$$s_1(y) = p(x_1 \mid y) \cdot p(y \mid y_0)$$

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y				
y'				
$\vdots$				
y^{last}				

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y	$s_1(y)$			
y'	$s_1(y')$			
$\vdots$				
y^{last}	$s_1(y^{last})$			

$$s_1(y) = p(x_1 \mid y) \cdot p(y \mid y_0)$$

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y	$s_1(y)$	$s_2(y)$		
y'	$s_1(y')$	$s_2(y')$		
$\vdots$				
y^{last}	$s_1(y^{last})$	$s_2(y^{last})$		

$$s_i(y) = p(x_i \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{i-1}(y')}$$

Viterbi Procedure (Part I: Prefix Scores)

	x_1	x_2	$\ldots$	x_ℓ
y	$s_1(y)$	$s_2(y)$		$s_\ell(y)$
y'	$s_1(y')$	$s_2(y')$		$s_\ell(y')$
$\vdots$				
y^{last}	$s_1(y^{last})$	$s_2(y^{last})$		$s_\ell(y^{last})$

$$s_\ell(y) = p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

Claim: $\max_{y \in \mathcal{L}} s_\ell(y) = \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\boldsymbol{x}, \boldsymbol{y})$

Claim: $\max_{y \in \mathcal{L}} s_\ell(y) = \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\boldsymbol{x}, \boldsymbol{y})$

$$\max_{y \in \mathcal{L}} s_\ell(y) = \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

Claim: $\max_{y \in \mathcal{L}} s_\ell(y) = \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\boldsymbol{x}, \boldsymbol{y})$

$$\max_{y \in \mathcal{L}} s_\ell(y) = \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$= \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{p(x_{\ell-1} \mid y') \cdot \max_{y'' \in \mathcal{L}} p}$$

Claim: $\max_{y \in \mathcal{L}} s_\ell(y) = \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\boldsymbol{x}, \boldsymbol{y})$

$$\max_{y \in \mathcal{L}} s_\ell(y) = \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$= \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{p(x_{\ell-1} \mid y') \cdot \max_{y'' \in \mathcal{L}} p}$$

$$= \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot$$

$$\boxed{p(x_{\ell-1} \mid y') \cdot \max_{y'' \in \mathcal{L}} p(y' \mid y'') \cdot \boxed{p(x_{\ell-2} \mid y'') \cdot \max_{y''' \in \mathcal{L}} p(y'' \mid y''') \cdot}}$$

Claim: $\max_{y \in \mathcal{L}} s_\ell(y) = \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\boldsymbol{x}, \boldsymbol{y})$

$$\max_{y \in \mathcal{L}} s_\ell(y) = \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$= \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{p(x_{\ell-1} \mid y') \cdot \max_{y'' \in \mathcal{L}} p}$$

$$= \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot$$

$$\boxed{p(x_{\ell-1} \mid y') \cdot \max_{y'' \in \mathcal{L}} p(y' \mid y'') \cdot \boxed{p(x_{\ell-2} \mid y'') \cdot \max_{y''' \in \mathcal{L}} p(y'' \mid y''') \cdot}}$$

$$= \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\bigcirc \mid y_\ell) \cdot p(x_\ell \mid y_\ell) \cdot p(y_\ell \mid y_{\ell-1}) \cdot p(x_{\ell-1} \mid y_{\ell-1}) \cdot p($$

$$p(x_{\ell-2} \mid y_{\ell-2}) \cdots p(x_1 \mid y_1) \cdot p(y_1 \mid y_0)$$

Claim: $\max_{y \in \mathcal{L}} s_\ell(y) = \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\boldsymbol{x}, \boldsymbol{y})$

$$\max_{y \in \mathcal{L}} s_\ell(y) = \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$= \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{p(x_{\ell-1} \mid y') \cdot \max_{y'' \in \mathcal{L}} p}$$

$$= \max_{y \in \mathcal{L}} p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot$$

$$\boxed{p(x_{\ell-1} \mid y') \cdot \max_{y'' \in \mathcal{L}} p(y' \mid y'') \cdot \boxed{p(x_{\ell-2} \mid y'') \cdot \max_{y''' \in \mathcal{L}} p(y'' \mid y''') \cdot}}$$

$$= \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} p(\bigcirc \mid y_\ell) \cdot p(x_\ell \mid y_\ell) \cdot p(y_\ell \mid y_{\ell-1}) \cdot p(x_{\ell-1} \mid y_{\ell-1}) \cdot p($$

$$p(x_{\ell-2} \mid y_{\ell-2}) \cdots p(x_1 \mid y_1) \cdot p(y_1 \mid y_0)$$

$$= \max_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} \prod_{i=1}^{\ell+1} p(x_i \mid y_i) \cdot p(y_i \mid y_{i-1})$$

High-Level View of Viterbi

- The decision about Y_ℓ is a function of $y_{\ell-1}$, $\boldsymbol{x}$, and nothing else!

$$p(Y_\ell = y \mid \boldsymbol{x}, \boldsymbol{y}_{1:(\ell-1)}) = p\left(Y_\ell = y \,\middle|\, \begin{array}{l} X_\ell = x_\ell, \\ Y_{\ell-1} = y_{\ell-1}, \\ Y_{\ell+1} = \bigcirc \end{array}\right)$$

$$= \frac{p(Y_\ell = y, X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}{p(X_\ell = x_\ell, Y_{\ell-1} = y_{\ell-1}, Y_{\ell+1} = \bigcirc)}$$

$$\propto p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot p(y \mid y_{\ell-1})$$

- If, for each value of $y_{\ell-1}$, we knew the best $\boldsymbol{y}_{1:(\ell-1)}$, then picking y_ℓ would be easy.
- Idea: for each position i, calculate the score of the best label prefix $\boldsymbol{y}_{1:i}$ ending in each possible value for Y_i.
- With a little bookkeeping, we can then trace backwards and recover the best label sequence.

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y				
y'				
$\vdots$				
y^{last}				

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y	$s_1(y)$ $b_1(y)$			
y'	$s_1(y')$ $b_1(y')$			
$\vdots$				
y^{last}	$s_1(y^{last})$ $b_1(y^{last})$			

$$s_1(y) = p(x_1 \mid y) \cdot p(y \mid y_0)$$
$$b_1(y) = y_0$$

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\ldots$	x_ℓ
y	$s_1(y)$ $b_1(y)$	$s_2(y)$ $b_2(y)$		
y'	$s_1(y')$ $b_1(y')$	$s_2(y')$ $b_2(y')$		
$\vdots$				
y^{last}	$s_1(y^{last})$ $b_1(y^{last})$	$s_2(y^{last})$ $b_2(y^{last})$		

$$s_i(y) = p(x_i \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{i-1}(y')}$$

$$b_i(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} p(y \mid y') \cdot s_{i-1}(y')$$

Viterbi Procedure (Part I: Prefix Scores and Backpointers)

	x_1	x_2	$\dots$	x_ℓ
y	$s_1(y)$ $b_1(y)$	$s_2(y)$ $b_2(y)$		$s_\ell(y)$ $b_\ell(y)$
y'	$s_1(y')$ $b_1(y')$	$s_2(y')$ $b_2(y')$		$s_\ell(y')$ $b_\ell(y')$
$\vdots$				
y^{last}	$s_1(y^{last})$ $b_1(y^{last})$	$s_2(y^{last})$ $b_2(y^{last})$		$s_\ell(y^{last})$ $b_\ell(y^{last})$

$$s_\ell(y) = p(\bigcirc \mid y) \cdot p(x_\ell \mid y) \cdot \max_{y' \in \mathcal{L}} p(y \mid y') \cdot \boxed{s_{\ell-1}(y')}$$

$$b_\ell(y) = \operatorname*{argmax}_{y' \in \mathcal{L}} p(y \mid y') \cdot s_{\ell-1}(y')$$

Full Viterbi Procedure

Input: $\boldsymbol{x}$, $p(X_i \mid Y_i)$, $p(Y_{i+1} \mid Y_i)$

Output: $\hat{\boldsymbol{y}}$

1. For $i \in \langle 1, \ldots, \ell \rangle$:
   - Solve for $s_i(*)$ and $b_i(*)$.
     - Special base case for $i = 1$ to handle start state y_0 (no max)
     - General recurrence for $i \in \langle 2, \ldots, \ell - 1 \rangle$
     - Special case for $i = \ell$ to handle stopping probability
2. $\hat{y}_\ell \leftarrow \operatorname*{argmax}_{y \in \mathcal{L}} s_\ell(y)$
3. For $i \in \langle \ell, \ldots, 1 \rangle$:
   - $\hat{y}_{i-1} \leftarrow b(y_i)$

Readings and Reminders

- Collins (2011), which has somewhat different notation; Jurafsky and Martin (2016)
- Quiz coming over the weekend!

References I

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34(1–3):211–231, 1999. URL `http://people.csail.mit.edu/mcollins/6864/slides/bikel.pdf`.

Kenneth W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ANLP*, 1988.

Michael Collins. Tagging with hidden Markov models, 2011. URL `http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/hmms.pdf`.

John M. Conroy and Dianne P. O'Leary. Text summarization via hidden Markov models. In *Proc. of SIGIR*, 2001.

Daniel Jurafsky and James H. Martin. Part-of-speech tagging (draft chapter), 2016. URL `https://web.stanford.edu/~jurafsky/slp3/10.pdf`.

Mark D. Kernighan, Kenneth W. Church, and William A. Gale. A spelling correction program based on a noisy channel model. In *Proc. of COLING*, 1990.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. In *Proc. of COLING*, 1996.

Extras

Something In Between Local and HMM

Slightly more generally, define features of adjacent labels in context: $\boldsymbol{\phi}(\boldsymbol{x}, i, y, y')$.

Features can depend on *any words at all*, just like in the local classifier.

Local Pairwise Classifier

$$(\hat{y}_i, \hat{y}_{i+1}) = \underset{y,y' \in \mathcal{L}}{\operatorname{argmax}} \, \mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y, y')$$

Local Pairwise Classifier

$$(\hat{y}_i, \hat{y}_{i+1}) = \operatorname*{argmax}_{y,y' \in \mathcal{L}} \mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y, y')$$

The problem is with disagreements: what if the $Y_{1:2}$ prediction and the $Y_{2:3}$ prediction do not agree about Y_2?

Even More Powerful: "Global" Prediction

As with the pairwise model, define features of adjacent labeled words in context: $\boldsymbol{\phi}(\boldsymbol{x}, i, y, y')$

"Structured" classifer/predictor:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathcal{L}^{\ell+1}}{\operatorname{argmax}} \sum_{i=1}^{\ell+1} \mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y_i, y_{i-1})$$

$$\overset{\text{HMM}}{=} \underset{\boldsymbol{y} \in \mathcal{L}^{\ell+1}}{\operatorname{argmax}} \log \left(\prod_{i=1}^{\ell+1} p(x_i \mid y_i) \cdot p(y_i \mid y_{i-1}) \right)$$

Even More Powerful: "Global" Prediction

As with the pairwise model, define features of adjacent labeled words in context: $\boldsymbol{\phi}(\boldsymbol{x}, i, y, y')$

"Structured" classifer/predictor:

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} \sum_{i=1}^{\ell+1} \mathbf{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}, i, y_i, y_{i-1})$$

$$\overset{\text{HMM}}{=} \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{L}^{\ell+1}} \log \left(\prod_{i=1}^{\ell+1} p(x_i \mid y_i) \cdot p(y_i \mid y_{i-1}) \right)$$

This is a fundamentally different kind of problem, demanding new:

- predicting ("decoding") algorithms
- training algorithms (to be discussed later)

Separating Two Ideas

HMMs are a specific probabilistic model for pairs of sequences $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

They are *also* the simplest example of a structured predictor: a collection of classifiers whose decisions depend on each other.